Title
Driver-Shift Design for Single-Hub Transit Systems under Uncertainty

Author
Munoz, Juan C.

Publication Date
2002-09-01

Driver-Shift Design for Single-Hub Transit Systems under Uncertainty

by

Juan Carlos Muñoz

Licenciado (Pontificia Universidad Católica de Chile) 1994

Magister (Pontificia Universidad Católica de Chile) 1996

M.S. (University of California, Berkeley) 2001

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering: Civil and Environmental Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Carlos Daganzo, Chair
Alper Atamtürk
Elizabeth Deakin
Samer Madanat

Fall 2002

The dissertation of Juan Carlos Muñoz is approved:

Chair Date

Date

Date

Date

University of California, Berkeley

Fall 2002

Driver-Shift Design for Single-Hub Transit Systems under Uncertainty

by

Juan Carlos Muñoz

Abstract

Driver-Shift Design for Single-Hub Transit Systems under Uncertainty

by

Juan Carlos Muñoz

Doctor of Philosophy in Civil and Environmental Engineering

University of California, Berkeley

Drivers account for up to 80% of the operational cost of transit agencies. This dissertation provides a method for improving the productivity of this workforce by introducing flexible contracts. Under these contracts drivers do not work the same number of hours every *day*. However, the number of days and hours worked every *week* are kept constant. These contracts allow the agency to find a better match between the number of drivers needed and hired. Since people's preferences vary, introducing this flexibility should benefit both the agency and the drivers.

Driver contracts are offered simultaneously to all drivers. Because drivers may be absent to work unexpectedly, these contracts must be designed to overstaff the system. However, current methodologies to determine these contracts fail to incorporate this uncertainty. There is also a lack of understanding as to how parameters of the problem such as driver wage, and the likelihood of absenteeism will impact the total cost. This dissertation aims to fill this gap. The methodology introduced here considers absenteeism, and provides cost sensitivities.

Further cost savings can be achieved if these contracts are synchronized with the trips operated by the agency. To achieve this goal, this dissertation extends the contract scheduling work and investigates the opposite problem where contracts are given and timetables must be identified. Two different methodologies are used to study this timetable design problem: continuum approximation and numerical optimization. The continuum approximation work incorporates uncertainties and provides cost sensitivities. The numerical optimization approach is used to validate the approximated results. In the latter work cuts are designed to solve an otherwise intractable problem. These cuts are also valid for the production *lot-sizing problem.*

Professor Carlos F. Daganzo, Chair

A Paula, mi princesa.

Acknowledgements

Work under the supervision of professor Carlos Daganzo has been a privilege. I will miss his dedicated guidance and enlightened comments. He has the outstanding capability of simultaneously understanding the big picture and the details. Every time we talked I left with a new angle on the problem, a cleaner idea of my research and fewer mistakes.

I want to thank professor Alper Atamtürk for the time he dedicated to this research and his guidance through combinatorial optimization. The rigor and determination with which he approaches his work are an inspiration to me. I also want to thank professors Betty Deakin, Samer Madanat, Michael Cassidy and Gordon Newell for their support on this research and through my years at Berkeley.

The students at Berkeley have been an excellent source of criticism and suggestions for this research. I want to thank specially Alex Lago, Anne Goodchild, Jorge Laval, Yuwei Li, and Karen Smilowitz.

This research has strong links with real transit systems. I want to thank the information provided by Gary Ward and Nacy Skowbo about ACTransit, Enrique Cabrera about Chilean passenger demand and by professors Avishai Ceder and Jarred Walker. Also, Ed Rothberg from ILOG helped me with cut management implementation in CPLEX 8.1.

I am grateful for the financial support of the Pontificia Universidad Católica de Chile which I will join soon, the Fulbright Program, the Ministerio de Planificación

Nacional de Chile and the University of California Transportation Center.

I want to thank my parents for all their love, and my wife Paula and our children Isabel, Ignacio and Joaquín for their patience and care during this adventure. Finally, I want to thank Juan de Dios Ortúzar for insisting that Berkeley was the best place for me.

Contents

List of Figures

Chapter 1

Introduction

1.1 The Problem

Transit agencies all over the world are under permanent pressure to reduce costs, while keeping high levels of service. Since labor cost is a major component of agencies budgets (in the U.S., around 75% of operational costs), improving the efficient use of drivers has become a priority for many transit providers.

Some inefficiencies in the use of labor stem from two important sources: driver contract rigidity and high absenteeism. Contracts, negotiated with strong unions, generally rule out driver part-time use (i.e., transit agencies are forced to hire drivers for full 8-hour workday). As a result, driver schedules can not be adapted to the high fluctuations of demand during the day, and agencies are doomed to use excessive work-shifts with significant driver idle hours during low-demand midday periods. At the same time, to cope with high absenteeism, agencies resort to *regular overstaffing* (i.e., regular drivers are hired in a larger number than needed) or *overtime* (i.e., the hired drivers are offered extra hour work at a premium wage), both of which result in substantial cost burdens. Identifying the adequate amount of overstaffing and overtime is a delicate decision overlooked by the current state of the practice.

This dissertation focuses on these inefficiencies and proposes innovative ways to deal with them more effectively. First, new flexible labor contract schemes are explored. These contracts are designed to benefit agencies, drivers and unions, because they combine the possibility of using drivers in part-time day shifts while still preserving all drivers to work full 40-hour weeks. The art of designing these contracts (i.e., when and for how long a driver should work) is referred as *driver-shift design*. Secondly, an efficient procedure is proposed to determine adequate driver-hiring levels to cope with absenteeism. This problem (i.e., how many drivers should be hired and on which contract) is referred as the *shift filling problem*. The combination of both problems is denoted as the *shift design problem*.

Agencies must also decide on the types of services to be offered (i.e., routes, timetables, vehicle requirements). Generally, the services are designed independently of labor requirements and contracts are defined subsequently to cover the planned timetables. However, additional labor efficiency could be achieved if timetables and driver shifts where jointly designed. This dissertation takes a step in that direction. This joint problem is referred as the *shift-timetable design problem*.

The design of timetables and driver contracts is essentially a planning process, since both timetables and contracts are fixed over the medium-long term (i.e., several months). To assess the benefits of any new design strategy, the total cost must be estimated over the planning horizon taking in account that the operational conditions may change day-to-day due to uncertainties. It is then critical to determine the cost impact of the different key parameters and decision variables. However, most agencies design their systems using deterministic optimization tools ill suited for planning under uncertainty. In this research, new planning tools are proposed that can correct these problems.

The following two sections of this chapter review the most relevant works and how transit agencies currently take these decisions. The last section states the main contributions of this dissertation.

Chapter 2 looks at the *shift design problem* in which all characteristics of the system are given except the driver contracts (number of drivers to hire, the shifts that each of these drivers will work and the trips that will be assigned to each of them). Flexible shift sequences are introduced and their impacts estimated. The deterministic problem without absenteeism is examined first. Later in the chapter, absenteeism is incorporated to the problem.

The results of chapter 2 provide a good platform to study the benefits of jointly designing driver shifts and timetables. First, chapter 3 looks at the complementary timetable problem: driver shifts are assumed known while timetables need to be determined. As in the previous chapter, the deterministic problem is tackled first and the stochastic later. At the end of chapter 3 the joint shift-timetable design problem is discussed with the help of some examples. In chapter 4 a methodology to solve the deterministic timetable design problem exactly using combinatorial optimization is provided. Conclusions are presented in chapter 5.

1.2 State of the Art

1.2.1 Related works

The process suggested in the literature to design transit systems is inherited from the airline industry. Currently in the airline industry a sequence of deterministic problems is followed in which the output of one problem serves as input of the following one. Figure 1.1 taken from [18] displays the sequential approach used for transit systems.

In this sequential approach, timetables (starting and ending times) for each line are obtained after the desired frequencies are stated. The result of this process is a set of chronological trips, each requiring a vehicle and a driver. In the sequential approach, each trip is assigned to a vehicle as determined by a *single depot vehicle scheduling problem* (SDVSP). In this problem a set of trips with given starting and ending time need to be covered by a set of vehicles so that every trip is assigned to a vehicle and each vehicle performs a feasible sequence of trips (the trip time between

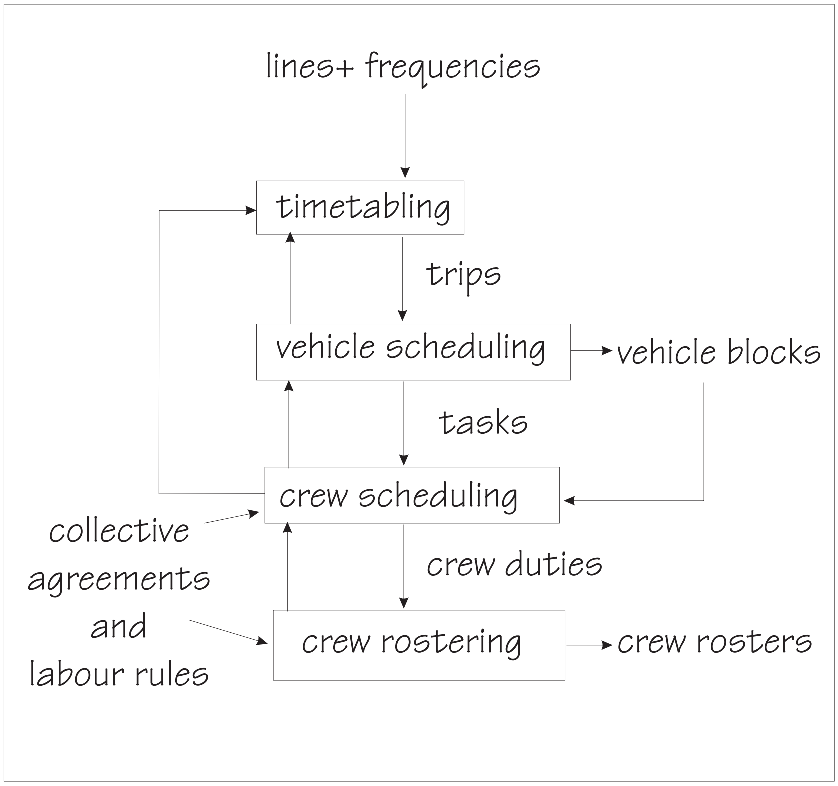

Figure 1.1: Sequential approach to define transit operations. Source: [18]

any pair of locations is also given so vehicles can be repositioned). The SDVSP can be solved in polynomial time. For a detailed overview of the literature in the SDVSP see [14].

Once the trips have been assigned to vehicles, the trips are divided into tasks that must be worked by a single driver[1]. Tasks start and end at *relief* points of the network where vehicles can exchange drivers. The problem of assigning drivers to tasks is called the *crew scheduling problem* (CSP). A CSP is solved for each day of the planning horizon. In the CSP, a minimum cost set of duties (sequence of

[1]Often called crew since the terminology was also inherited from the airline industry

tasks by a single driver) must be determined so that all tasks are assigned to a duty and each duty can be performed by a driver (in the transit industry "duties" are called "runs"). The possibility of drivers doing overtime at a higher salary is usually considered in a CSP. Driver constraints like maximum working time without a break, minimum break duration and maximum working time are typically added. Although CSP looks similar to the SDVSP in structure, it is much more complicated because in the SDVSP duties are constrained to full tasks. The CSP has been shown to be NP-hard (see [17]).

The three most widely used methods to solve the CSP are([27]): the runcutting heuristics, the HASTUS method and the set covering method. The first two are heuristic approaches while the third can yield an optimal solution. In the runcutting heuristic, schedules are built choosing tasks that form feasible duties. Then, the heuristic tries to improve the solution by cutting and exchanging tasks among duties (see [36]). The HASTUS method first solves the linear programming relaxation and then performs local improvement techniques. Finally tasks are combined to form a feasible (see [29]). Finally, the set covering method formulates the problem using a set covering model in which each variable represents a feasible workday to be worked by a driver covering its duties. Some of the papers following this approach use a reduced number of feasible workdays (see [30] and [36]) while others implicitly consider all of them using a column generation approach (see [12], [13], [23] and [33]).

Once the driver duties over the planning horizon have been determined, duties are assigned to specific drivers. At this stage, called the rostering process, other constraints are considered like rest periods, holidays, training, etc. This problem has attracted a lot of attention in the last decade especially for the airline industry (see [6], [11] and [19])

Notice that driver-shift design is not considered in this sequential process. In the case of airlines this is to be expected since trips and workdays are of similar length so crews tend to work few (sometimes only one) trips per workday. However, for transit systems where drivers work several trips per day and demand has predictable peak and off-peak periods, driver-shift design plays an important role in the system performance.

The approach for transit systems is inherited from the airline industry, where vehicles are optimized before drivers (crews in this case) since vehicle related costs have a bigger impact on the total budget. In the case of transit systems driver related costs are the most significant. Therefore, transit design systems may benefit from optimizing drivers first and should do it taking full advantage of driver's flexibility.

This sequential approach to design a transportation system has other important limitations. Performance improvements could be obtained if the problem were solved simultaneously instead but the complexity of the problem makes it a challenge. During the last decades, the simultaneous vehicle and crew scheduling problem has been studied based on heuristic procedures ([4], [16] and [28]). In [18] the SDVSP is formulated and solved using column generation applied to Lagrangean relaxations of the problem. The authors claim that the benefits of solving the vehicle and crew scheduling problems simultaneously over a sequential approach are greatest if changeovers (driver changing vehicles during a duty) are not allowed. However, according to [7] no publication has explored the problem of simultaneously designing timetables and crew schedules, . All this is not surprising because the complexity of real networks makes the problem very hard (a special case of this problem was shown to be NP-hard, see [17]).

Another drawback of the sequential approach is that it assumes a deterministic system. However, uncertainties are present at all operation levels (e.g. driver absenteeism, passenger demand, roundtrip durations, vehicle failure, etc.). Absenteeism appears to be very typical of transit systems and very disruptive too. In [3] it is concluded that transit operators in the U.S. were (unexpectedly) absent 11.9% of the work days compared with 3% for all U.S. employees. This study also concludes that absenteeism consumed more than 25% of the federal subsidy for transit and was increasing rapidly. According to [35], “there is a strong belief in the transit industry that the ready availability of overtime influences the amount of absence occurring”. However, in [31] the relation between absenteeism and overtime is studied, concluding that reducing overtime will not necessarily reduce absence. It is suggested instead to develop a monitoring system to identify problematic drivers and a disciplinary system to improve attendance.

1.2.2 Methodologies

Most papers regarding crew scheduling problems use a numerical optimization approach heavily relying in integer programming tools. Although this might be a good approach for a short-term operation where the problem's parameters are fixed, it is not as suitable for planning purposes since uncertainties are ignored.

Thus, the solution methodologies must incorporate uncertainty in the variables and the parameters. A successful approach for these type of problems is to treat some discrete parameters and variables as continuous. This analytic approach can yield near optimal solutions expressed in closed form using few key parameters. Continuous approximations have been successfully applied to freight distribution, scheduling and location problems (see [9], [10], [15], [20], [26] and [32]). For a comprehensive overview of recent works on freight distribution systems using this approach see [22].

In this dissertation some discrete phenomena are modelled using continuum approximations. This approach yields a total cost function that can be realistically represented with only a few parameters, allowing the use of optimization techniques and sensitivity analysis.

Continuous solutions must be again translated into discrete decisions for implementation. In the real world, the shifts are made of non-continuum people and the shifts must start and end at a discrete and finite set of instants along the day (e.g. every hour, or half an hour). Therefore, executing the optimal or near-optimal solution requires accommodating them to fit these real world constraints. Heuristics are suggested to build this link.

As a final step, the precision of the cost obtained with the continuum approximation is tested. Since the deterministic versions of the problems in this dissertation can be modelled using mathematical programming techniques, bounds on the optimal solutions are searched using numerical optimization. For a comprehensive and thorough treatment of combinatorial optimization techniques, see [24].

1.3 State of the Practice

This section briefly explains current practices for determining timetables and driver scheduling in the transit industry. Since these processes vary worldwide the focus is on the Alameda-Contra Costa Transit System (ACT).

1.3.1 Alameda-Contra Costa transit system

Like most transit agencies in major U.S. cities, ACT designs its system based on the sequential approach described in section 1.2. Once ACT defines the routes that will be offered, the timetable per route is determined by the agency to satisfy certain levels of service. These timetables follow the demand and therefore frequencies are higher at peak hours. Although most routes offered by ACT operate for approximately 20 hours a day, some provide ‘round the clock’ service.

The timetable is composed of trips that need a bus and a driver. The next step is to group trips to be run by the same bus. These groups are called blocks. Blocks can be of any length (as long as a full work-day). Blocks start and end in garages. Since a block may have several drivers assigned, its length is limited only by the fuel capacity of the bus. In the case of ACT, buses can operate continuously for up to 300 miles.

The daily blocks are then divided into tasks that can be operated by a single person. The set of tasks assigned to a driver is called a *run* by ACT (called a "duty" in the previous section). The process of dividing blocks into runs is called *runcutting*. Runs can start or end at a garage or at a *relief point* at a street intersection. Runs can be continuous or split. On a split run, a driver works a certain number of hours, leaves the work site for an extended period of time, and reports back to work later in the same day.

ACT considers a *shift* to be the contracted work hours for a driver. Runs are constrained by the length of a shift. Like runs, shifts can be continuous or split. Shifts must follow the specifications of the amalgamated union contract and the current legislation. ACT guarantees each driver 8 working hours per day. Although most runs last around 8 hours per day, runs of 9, 10 or more hours are common in ACT and in the industry. Work over eight is considered overtime and is typically paid 1.5 times the regular pay. Often, overtime is perceived as an entitlement, and unions sometimes object to streamlined runcuts that reduce overtime ([34]). Current legislation forbids the number of driving hours (platform time) per shift to exceed

10 hours and the number of working hours (spread time) to exceed 15 hours (ACT standard is only 13 hours for spread).

Once runs have been defined over the planning period, the *rostering* process begins. In this process, runs on different days are grouped in sets that a single person can work (considering days off, vacations, training, etc.). The result is a list of contracts (although agencies do not call them contracts) from which the workers choose in seniority order. This process is called a "sign-up". Notice that the contracts define not only when the driver should be working but also what the driver should do.

Drivers assigned to certain trips can not be forced to work different ones during the term of the sign-up. To deal with absenteeism, based on ACT's Chief Dispatcher experience, ACT hires an additional 25% work force as extra-board drivers. These drivers are regular, full-time employees, and unionized, but their work schedule fluctuates on a daily basis. Extra board drivers sign for either the day or the night board. According with ACT's Collective Bargaining Agreement:

"Day Extra Board operators shall not be scheduled to start work later than 12:00 noon. Night Extra Board operators shall not be scheduled to start work earlier than 12:00noon. The only exception shall be to avoid cancellation in service. A Night Extra Board bus driver working one of his/her regularly assigned work days, shall receive payment of time and one half for all work performed before 12:00 noon. All work that is out after 8:00p.m. is designated as night work."

Likewise, if a day Extra Board driver on a regular work day is assigned a run that works past 8:30pm, he/she is paid time and one-half for the work performed after 8:30pm. Some tasks that these drivers do can be predicted with a day of anticipation. Every day before noon some of these drivers are assigned to jobs they will do on the next day. The remaining extra-board drivers are called *point drivers* who will be assigned to jobs as needed (within the parameters of the collective bargaining agreement)[2]. The times that point drivers are requested to arrive for work are determined based on ACT's Chief Dispatcher experience.

The design procedure followed by ACT (and by many other agencies in the US) has many drawbacks. First, to deal with absenteeism agencies must correct the output of the process, by adding 25% of drivers to the deterministic assignment based on experience. Since uncertainties are inherent and quite significant to all transit systems, a more adequate planning tool is needed.

Second, extra board drivers learn of their work schedule with at most one day advance notice [3]. This is clearly undesirable and therefore most extra board shifts

[2]This assignment is done following a list. If some drivers are not assigned to trips on one day, they are the first to be considered on the next day. In this way all drivers are equally likely, in the long term, to be assigned a task within a short or long call.

[3]The following example was provided by ACT's Chief Dispatcher: "A driver has a point shift at 6am, this person sits for 3 hours, until 9am, at which point the Dispatcher asks her/him to go home and report again at 12:30pm. The driver comes back, and sit until 2pm, when a trip from 2:30pm to 6:30pm opens up. The Dispatcher gives the driver the assignment, which (s)he must accept since it is within the 13 hours spread for the day".

are usually the last to be chosen by drivers.

Third, the seniority order in which shifts are chosen by drivers creates a problem since the most experienced drivers tend to choose the shifts that are more pleasant (and easier) to drive leaving the harder shifts to the newcomers.

This dissertation suggests more flexible contracts based on guaranteeing drivers 40 hours and two days off a week instead of 8 hours a day. Additionally, a different strategy for assigning shifts to drivers is provided. In this strategy no distinction is made between regular drivers and extra board. All of them will know their schedule for the whole term of the sign-up but will not know in advance which trips they will work. This strategy also provides the agency some flexibility to assign drivers to trips according with driver experience.

The planning approach followed by agencies does not allow for analysis of the sensitivity to the parameters and variables on the total cost. For example, agencies do not have a clear understanding of how their costs would increase if wages were to rise by 10%, if drivers were to be absent more often or if peak frequencies were to be doubled. This dissertation provides tools that should help the agencies to answer these questions.

1.4 Summary of Contributions

This research provides the following contributions:

- Suggestion and evaluation of heterogeneous shift combinations. These combinations are constrained to 40 hours and two days off a week instead of the more traditional constraint of 8 hours a day.

- Development of a new strategy to allocate drivers to shifts. This strategy allows the agency some flexibility to take advantage of driver abilities and experience.

- Development of a planning tool for the deterministic and stochastic shift filling problem. The impact of the relevant variables and parameters in the agency's performance can be estimated with this tool.

- Development of a planning tool for the deterministic and stochastic timetable design problem. The impact of the relevant variables and parameters in the agency's performance can be estimated with this tool.

- Development of new cutting planes for the deterministic timetable design problem. Facet defining conditions are provided. These cuts are also valid for the production *lot-sizing problem* (see [2]).

- Development of a procedure to solve the timetable-shift design problem.

Chapter 2

Shift Design Problem

2.1 The Problem

Consider a transit agency that must redesign the contracts offered to its drivers. These new contracts will remain valid over a certain planning horizon (e.g. some months). Imagine that the rest of the operation (routes, timetables, fleet size, etc.) remains unchanged. This problem is the *shift design problem.* The problem can be decomposed in two subproblems: identifying attractive types of contracts for the agency and drivers, and finding which of those contracts should be offered to the drivers so that the expected cost over the planning horizon is minimized. The first problem is called here shift-driver design, while the second is referred as the shift filling problem. The solution of the shift filling problem depends on two aspects:

- the transit system (e.g. the timetable followed by each line, the length of the trips, the time drivers lose between two consecutive trips, etc.).

- the contract types that the agency and the drivers are willing to sign (referred in the introduction as driver-shift design).

In the next two sections these aspects are simplified for analysis. The deterministic shift filling problem is considered first. Later, absenteeism is included as a source of uncertainty.

2.2 Assumptions

2.2.1 The system

Two types of assumptions are made regarding the system. The first concerns the demand faced by the transit system and the consequent timetable offered. The second corresponds to network simplifications that make the shift design problem easier.

Timetable assumptions

Timetables are directly influenced by passenger demand. During high demand periods, frequencies should be higher too. This section analyzes passenger demand and the resulting timetables focusing on weekdays. Daily passenger demand for most transit systems worldwide is quite predictable in the medium term of one year. However, the transit design tools developed in the literature do not make any assumptions on the daily passenger demand. Although this looks like a strength of those methods, it may in fact be a weakness. To design the system the agency does not need a very powerful methodology that will bring a good solution for any possible input, but instead a tool that can assess the impact of the different parameters on the total cost. The simpler and more characterized the inputs, the easier it is to provide such a tool.

In large cities most people begin and end work within small time windows. Therefore transit systems usually face a double-peaked demand in which both peaks have a similar height (morning being usually higher than afternoon because the start of the work day is less scattered among workers) and are more than a typical work day length apart (usually around eight hours). Therefore, although three consecutive 8-hour continuous shifts can cover the whole work-day, no continuous shift (without overtime) can cover both peak periods. Figure 2.1 illustrates the demand profile that transit systems face. This particular profile corresponds to the average 10-minutes daily boardings on the AC Transit system over several weeks. It should be expected that most transit systems in major cities face a demand that can be adequately approximated by rescaling the demand in the figure. For example in Santiago de Chile (a city of 5 million inhabitants) the demand is similar. Figure 2.2a and 2.2b show the 15-minute boardings for the underground (Metro) and bus systems, respectively. However, in some cultures people go home for lunch. Thus, the noon-peak is much more noticeable and sometimes even taller than the evening peak period. Figure 2.3a and 2.3b show the same information for two middle size (around 1 million inhabitants) cities in Chile: Valparaíso and Concepción. For the purpose of this dissertation it is assumed that every weekday the demand profile faced by each line of the system is proportional to the one depicted in figure 2.1. However, a pool of different morning-noon peak ratios is considered when needed.

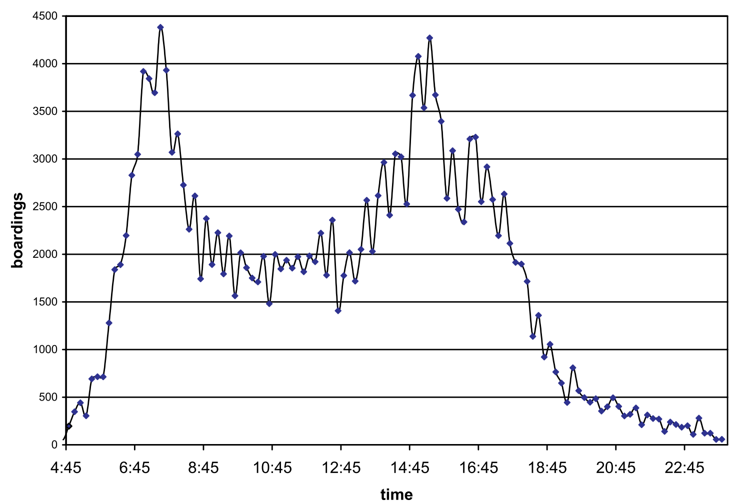

Figure 2.1: AC Transit daily 10 minutes boardings systemwide

Weekday demand is assumed to be double-peaked and is approximated with a smoother curve as shown in figure 2.4a. This curve is a piecewise cubic approximation of the real data. In the absence of capacity or driver constraints, if an agency wants to minimize passenger's waiting time expressed in cost units plus operation cost, vehicles should be dispatched at a rate proportional to the square root of the passenger arrival rate, see [25]. A timetable is derived following that rule assuming a single line with the boardings of figure 2.4a, a waiting cost over operational cost ratio of 5/12, and no further constraints on the scheduling process (e.g., vehicle ca-

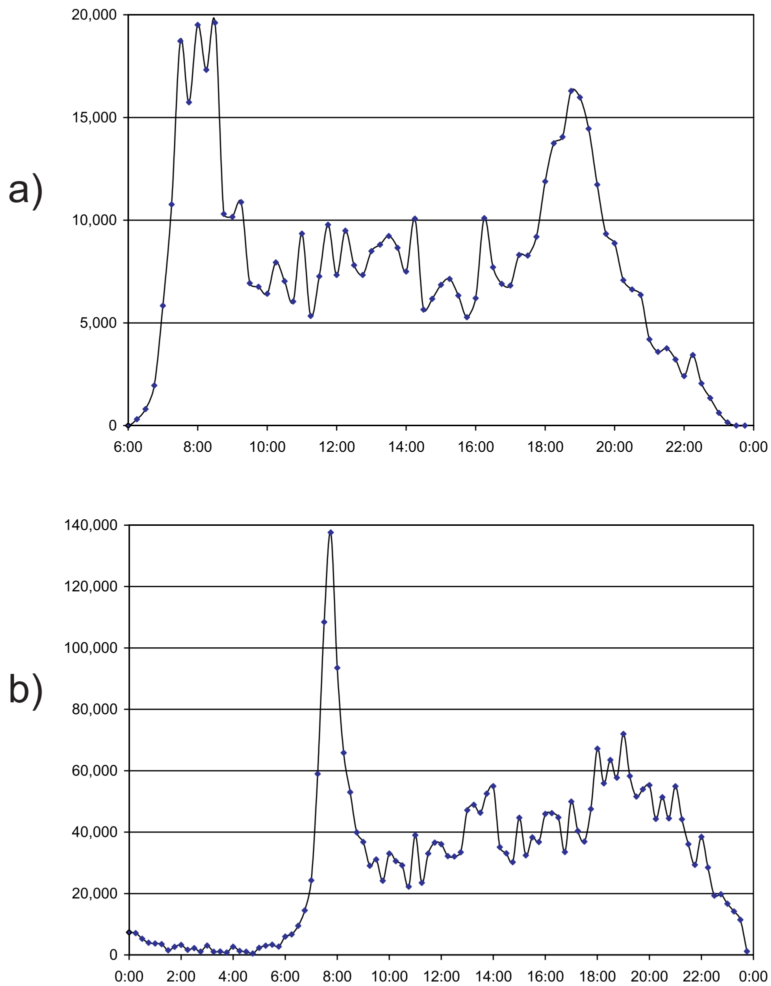

Figure 2.2: Daily boardings (every 15 minutes) in Santiago de Chile in 2001; a) Metro system; b) Bus system

pacity, driver availability or minimum frequency). As expected, the schedule (shown in figure 2.4b)is double-peaked, but not as spiky as the demand curve. This curve provides an illustration of a typical system timetable. However, in the real world some additional constraints may slightly distort this profile during peak and off-peak hours. During peak hours, some lines might face active capacity constraints while

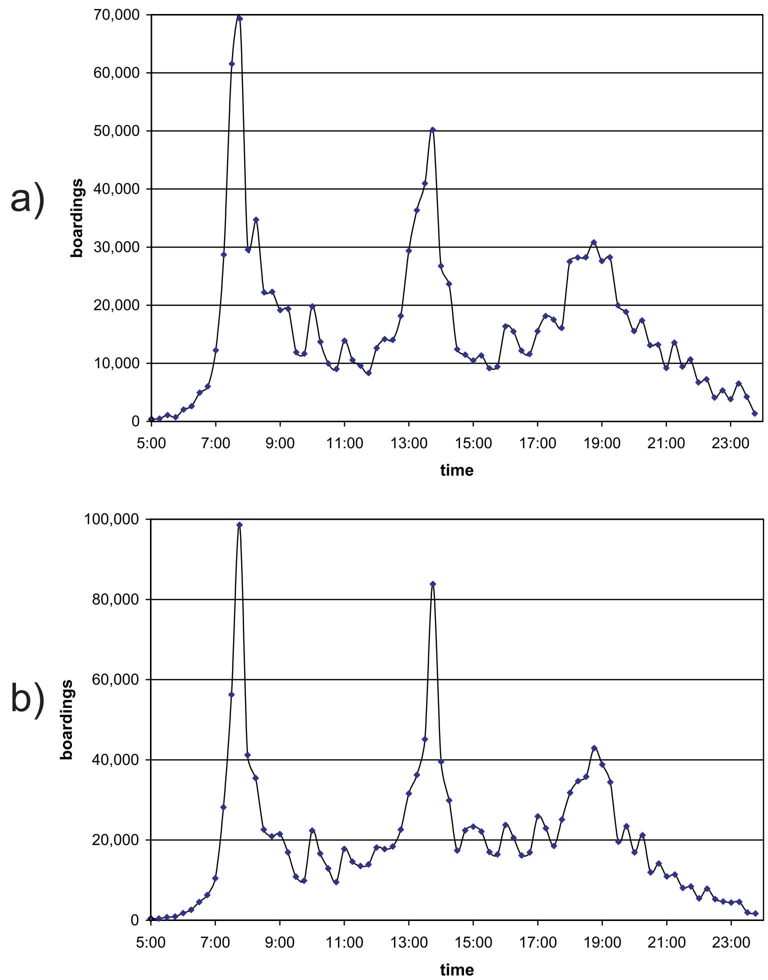

Figure 2.3: Daily boardings (every 15 minutes) in middle size cities; a) Valparaíso (Chile) in 1999; b) Concepción (Chile) in 1998

during off-peak hours (e.g. late evening and night) a minimum frequency often must be offered. In all cases the timetable profile on a weekday should be qualitatively similar to the one derived here. This dissertation assumes that on weekends agencies offer a flat frequency all day.

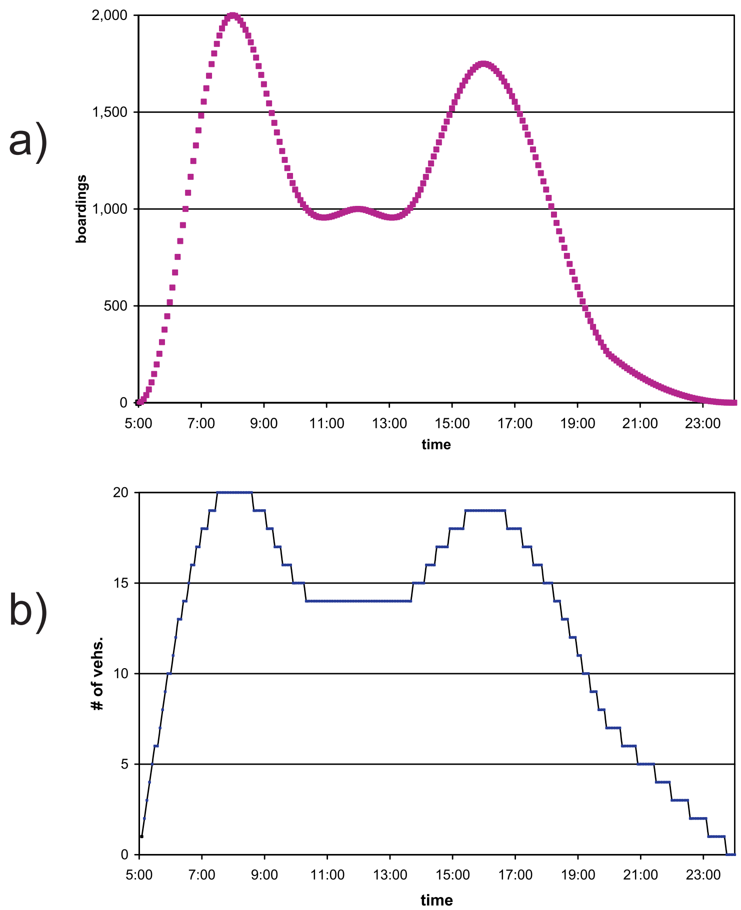

Figure 2.4: a)5-minutes boardings to the system; b) 5-minutes dispatching schedule

The network

To simplify the analysis, a particular network type is assumed. Although the system may operate many lines, each starts and ends at the same station (like a hub and spoke system, see figure 2.5). This allows drivers to transfer from one route to another without a penalty.

The *roundtrip duration* is defined as the time elapsed since a bus departs from the hub until it arrives back and its driver is ready to start a new trip. Vehicles face

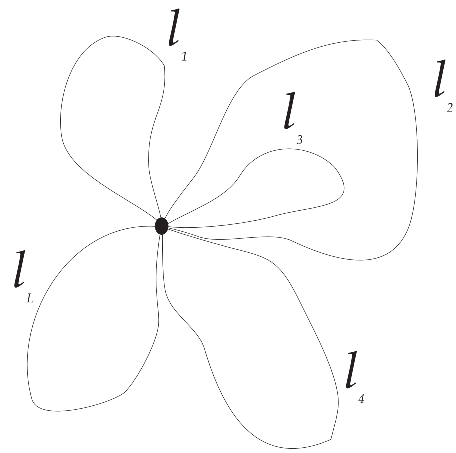

Figure 2.5: Hub and spoke transit system

no congestion on the roads and stay for a fixed time period at each stop (loading and unloading). Thus, the roundtrip duration of each line is assumed fixed.

In this dissertation no relief points but the hub are considered. As a result, trips can not be driven by more than one driver. Where appropriate the advantage of having relief points is highlighted. The number of started but unfinished trips in the system at any moment in time is called the *active tasks* at that time and denoted as $v(t)$.

2.2.2 Driver contracts

Before analyzing contracts some definitions are needed.

A *shift* consists of one or two time intervals to be operated by a driver on a single day with well defined start(s) and end(s). A one interval shift is called a continuous shift. A two interval shift is called a split shift. The intervals on any split shift can not overlap. One must be allocated along the morning peak hour and the other along the afternoon peak hour.

The intervals of a shift are classified by type, f, according to their length and whether they are a *daily* interval, a *morning* interval or an *afternoon* interval. Every morning interval must have a corresponding afternoon interval. For example work on Monday October 7 between 9:00 and 17:00 hours is classified as an $f = 8$ interval type ("daily" intervals are assumed as default). For a split shift example, consider working Monday October 7 between 7:00 and 10:00 hours and between 15:00 and 20:00 hours. The two intervals on this shift are classified as $f = 3m$ and $f = 5a$, respectively (m for morning and a for afternoon). Notice that interval types do not specify starting or ending times. The set of all allowable interval types is F.

An *interval type combination*, or *combination* for short, is a periodic ordered sequence of daily interval types [1] whose cycle is a number of weeks. The set of interval types assigned to any weekday for a whole cycle must be the same for all weekdays. Table 2.1 shows three combinations. The first combination has a single-week cycle, and consists of an eight-hour interval type five days a week. The second combination has a five-week cycle and consists of a nine-hour interval type four days a week, and a four-hour interval type one day a week. Note that all weekdays are alike. The third combination also has a five-week cycle and consists of an eight-hour interval type three days a week, and a four-hour morning and a four-hour evening

[1] Each day includes either an ordinary type or a morning-afternoon pair.

interval types twice a week. Again, as required, all weekday columns have the same set of interval types. Thus, in the long run, a driver working a combination will be assigned to the same number of intervals of each interval type on all weekdays. In this dissertation only combinations with one-week, five-week, and ten-week cycle will be considered. The set of all combinations considered by the agency is called the *combination menu* and denoted by S, while the set of all interval types in combination $s, s \in S$ is called $F(s)$. A combination with well-defined starting time for each of its intervals is called a *shift sequence*. The least common multiple of the cycles of the combinations in the combination menu is called the *menu cycle*.

Notice that since combinations are cyclic, they can be started at different phases of the cycle yielding similar but not identical combinations. If these combinations are offset by one or more weeks, they are called *sisters*. As an example, combination 2 in table 2.1 yields 5 different sister combinations. Thus, there are as many sister combinations as the size of the cycle (in weeks). Drivers working a certain combination will always be hired in batches of size equal to the cycle size (in weeks) so that each driver works a different sister of the combination. Then, the drivers of a batch work the same set of interval types all days (excluding weekends) and all weekdays are equal in terms of interval types. The batching restriction allows the problem to be decomposed on a day by day basis, reducing its complexity considerably.

In this research the following assignment process is assumed. The agency starts by defining the combination menu (this requires the agency to understand the

Comb	Monday	Tuesday	Wednesday	Thursday	Friday	Sat	Sun
1	8 hrs	8 hrs	8 hrs	8 hrs	8 hrs	-	-
2	4 hrs	9 hrs	9 hrs	9 hrs	9 hrs	-	-
	9 hrs	4 hrs	9 hrs	9 hrs	9 hrs	-	-
	9 hrs	9 hrs	4 hrs	9 hrs	9 hrs	-	-
	9 hrs	9 hrs	9 hrs	4 hrs	9 hrs	-	-
	9 hrs	9 hrs	9 hrs	9 hrs	4 hrs	-	-
3	4 hrs, 4 hrs	8 hrs	4 hrs, 4 hrs	8 hrs	8 hrs	-	-
	8 hrs	4 hrs, 4 hrs	8 hrs	4 hrs, 4 hrs	8 hrs	-	-
	8 hrs	8 hrs	4 hrs, 4 hrs	8 hrs	4 hrs, 4 hrs	-	-
	8 hrs	4 hrs, 4 hrs	8 hrs	4 hrs, 4 hrs	8 hrs	-	-
	4 hrs, 4 hrs	8 hrs	8 hrs	8 hrs	4 hrs, 4 hrs	-	-

Table 2.1: Illustration of three interval type combinations

drivers' and union's preferences and needs). Using this menu and the timetable of each line as inputs the agency identifies a list of intervals (with defined starting times) for every weekday. Each interval is associated with a particular combination. The intervals associated with a combination must be "feasible"; i.e., in the correct proportions for the combinations in question to be formed. This output determines the number of combinations, by type, that will cover the timetable, and hence the number of drivers that will work each combination. A methodology to identify an optimum list of feasible intervals is described later in this chapter.

To complete the process the agency must group the intervals into feasible shift sequences in the order of each combination and its sisters. When split shifts are part of the combination, non-overlapping morning and afternoon intervals must be chosen. The morning and afternoon intervals associated with the same days in the combination should be sorted in chronological order and paired before the assignment.

Each of these sequences (plus its wage for regular and extra hours) defines a "contract". Notice that the contract does not specify which trips will be driven. The driver should be prepared to run any trip on any line during her/his workday. The contract must also define how long it will be valid. This period must be an integer multiple of the menu cycle. Finally, these contracts are posted for the drivers to choose. Once a driver has been assigned to a contract, (s)he must work the hours specified. At the end of each day the driver must notify the agency if (s)he will be absent the following day. At the beginning of each day the agency notifies all drivers which specific trips they will have to work on that day and if overtime and/or extra jobs are being offered to her/him. The driver must accept or reject the extra work at the beginning of the day. If a trip is rejected by all drivers, a tripper is hired to work it.

Shift combinations that workers and agencies might like

A complete review of combinations used in different industries around the world is provided in [8]. The advantages and disadvantages of each of the different combinations for the firms and for the workers are described (although none of the cases studied in this reference are taken from the transit industry). As correctly expressed in [8], workloads should balance business needs, employee desires and health and safety requirements. In many sectors of the economy, this is arranged by operators working 8-hour a day, 5-days a week. However in some other sectors this is not a desirable solution. For instance, in [8], the 24 hour business is considered. In this case some operators must work undesirable shifts (e.g. night shifts). The author claims that, according with his experience, substantial gains can be obtained by looking at less orthodox combinations. A transit system might not necessarily operate 24 hours a day but since it must provide weekday service according with a double-peaked demand, traditional 8-hour a day, 5-days a week proves to be very inefficient for the agency. The goal of the agency in this respect is clear: to have at every instant in time the right number of workers (not over- or understaffed) at a minimum cost. This means more drivers on peak hours. Thus, transit firms have tried different shift alternatives: accommodating part time drivers during peak hours, splitting shifts, hiring trippers to work particular trips, extending current drivers contracts as overtime, or even outsourcing the peak services to a third party. Since these alternatives are not always attractive for drivers, higher wages are offered. In addition, unions concerns need to be considered. Unions prefer all drivers to be hired as full time workers to avoid drivers getting the unions' benefits but working considerably less (e.g. part time workers).

To deal with the weekday peaking problem transit agencies should take advantage of two characteristics of the system. First, service must be provided during weekends giving an extra degree of freedom that could be used to accommodate shifts during weekdays better. Second, all drivers are different and therefore have different preferences. Thus, a combination that is not attractive for someone might appeal to somebody else. From the agency's perspective, a combination menu with multiple combinations will fit the schedule considerably better than using just a single kind. Therefore, workforce diversity favors the agency and combination diversity favors the workforce. Although agencies have profited from the first of these two characteristic, the second has not been exploited. The next section explores heterogeneous combinations that may appeal to the agency, the drivers and the unions.

Heterogeneous combination examples

Traditionally, transit drivers expect to work a combination of 8-hour a day, 5-days a week (like combination 1 in table 2.1). Although this combination might be acceptable for most drivers, some drivers might prefer different ones. This section explores new combinations that still satisfy the following two constraints: forty hours of work and an average of two days off per week. For instance, consider the following three

Comb	Monday	Tuesday	Wednesday	Thursday	Friday	Sat	Sun
4	5 hrs	10 hrs	5 hrs	10 hrs	10 hrs	-	-
	10 hrs	5 hrs	10 hrs	5 hrs	10 hrs	-	-
	10 hrs	10 hrs	5 hrs	10 hrs	5 hrs	-	-
	10 hrs	5 hrs	10 hrs	5 hrs	10 hrs	-	-
	5 hrs	10 hrs	10 hrs	10 hrs	5 hrs	-	-
5	12 hrs	8 hrs	4 hrs	12 hrs	4 hrs	-	-
	4 hrs	12 hrs	8 hrs	4 hrs	12 hrs	-	-
	12 hrs	4 hrs	12 hrs	8 hrs	4 hrs	-	-
	4 hrs	12 hrs	4 hrs	12 hrs	8 hrs	-	-
	8 hrs	4 hrs	12 hrs	4 hrs	12 hrs	-	-

Table 2.2: Illustration of two non-weekend combinations

combinations:

- Four days working nine hours and one day working only four (like combination 2 in table 2.1).
- Three days working ten hours and two days working only five.
- Two days working twelve hours, one day working eight hours and two days working four hours.

Table 2.2 illustrates the last two combinations. These types of combinations are not standard currently because any work after 8 hours is considered overtime.

However, people's choices vary and many people might prefer to have a shorter day once a week instead of all equally long. A shorter work day on a Monday or a Friday could translate into a longer weekend for aa driver. On the other hand, a varied set of shift types helps the agency to match better its needs throughout the day. Therefore, the above heterogeneous combinations might become a win-win alternative.

The above combinations do not consider work during weekends. However, if this degree of freedom is added new combinations may be developed. Even though the combinations that are proposed next request some weekend work, they are still 40 hours and five days in average a week, and never request work on Sundays or on consecutive weekends. Consider the following two combinations where the driver works ten hours four days a week (no work on weekends) in odd weeks, while in even weeks the driver works:

- four hours two days, six hours two days and ten hours on a weekday and on Saturday, or

- two hours one day, four hours two days and ten hours two days and on Saturday.

Table 2.3 illustrates these combinations which offer a long weekend once every ten weeks.

It is not claimed here that everybody will prefer these combinations. However, an informal sampling of people has confirmed that a majority prefer less conventional

Rot	Monday	Tuesday	Wednesday	Thursday	Friday	Sat	Sun
6	10 hrs	10 hrs	10 hrs	10 hrs	-	-	-
	4 hrs	10 hrs	6 hrs	6 hrs	4 hrs	10 hrs	-
	10 hrs	10 hrs	10 hrs	-	10 hrs	-	-
	10 hrs	6 hrs	6 hrs	4 hrs	4 hrs	10 hrs	-
	10 hrs	10 hrs	-	10 hrs	10 hrs	-	-
	6 hrs	6 hrs	4 hrs	4 hrs	10 hrs	10 hrs	-
	10 hrs	-	10 hrs	10 hrs	10 hrs	-	-
	6 hrs	4 hrs	4 hrs	10 hrs	6 hrs	10 hrs	-
	-	10 hrs	10 hrs	10 hrs	10 hrs	-	-
	4 hrs	4 hrs	10 hrs	6 hrs	6 hrs	10 hrs	-
7	10 hrs	10 hrs	10 hrs	10 hrs	-	-	-
	2 hrs	10 hrs	4 hrs	4 hrs	10 hrs	10 hrs	-
	10 hrs	10 hrs	10 hrs	-	10 hrs	-	-
	10 hrs	4 hrs	4 hrs	10 hrs	2 hrs	10 hrs	-
	10 hrs	10 hrs	-	10 hrs	10 hrs	-	-
	4 hrs	4 hrs	10 hrs	2 hrs	10 hrs	10 hrs	-
	10 hrs	-	10 hrs	10 hrs	10 hrs	-	-
	4 hrs	10 hrs	2 hrs	10 hrs	4 hrs	10 hrs	-
	-	10 hrs	10 hrs	10 hrs	10 hrs	-	-
	10 hrs	2 hrs	10 hrs	4 hrs	4 hrs	10 hrs	-

Table 2.3: Illustration of two combinations including some weekend work

combinations. The agency can take advantage of the diversity by offering a pool of combinations. Again, people's diversity should be considered when combinations are defined since it plays in everyone's favor.

Some of the described combinations may not satisfy safety and health regulations, such as twelve-hour shifts. These concerns should be considered before determining the combination menu to be offered. Some rest period should be included as part of the trip time. Long intervals should include breaks of fixed length. All transit systems must be designed overstaffed to account for driver absenteeism, therefore on average drivers will experience rest periods longer than these breaks. Therefore, the rest period added to every trip can be set as the minimum required for safety. To reduce the cumulation of work, long interval types should be scheduled as far away as possible from each other within the week as in combinations 2 to 7 in the tables.

These concerns are fully recognized in current practice. Currently, drivers are being offered contracts for shifts of eight hours plus two or three hours of overtime. Since overtime is paid at higher rates, these combinations are the most valued by drivers and therefore chosen by senior drivers. Most senior drivers are not necessarily the drivers that should be operating long shifts but this is the way many systems work today.

2.3 The Deterministic Problem

The inputs of the shift filling problem are the timetables, the network characteristics and the combination menu. The outputs are the (feasible) daily intervals and their posterior grouping into shifts and sequences (according with combinations in the menu). The intervals must cover all the trips in the timetable at minimum cost.

Since the network is hub-and-spoke, drivers can work trips on different lines sequentially without a lag. Therefore, trips can be modelled by their starting times and roundtrip durations [2].

Notice that passenger demand has no impact on this problem since the timetable and waiting times are fixed (it is assumed that all trips are run on time). Additionally, the operational costs are fixed since the trips are fixed. Therefore, the only variable costs in this problem are driver wages.

The problem is formulated using a set-covering model where a set of characteristics (in this case the daily trips) needs to be covered by a number of activities (the work intervals). The objective is to determine the least costly combination of activities that collectively possess (cover) each characteristic at least once.

In the formulation that follows, a big source of intervals is considered. Some of these intervals will be chosen and some will not. Each chosen interval is associated with an interval type of specific length and a combination. Each trip must be assigned to one of these intervals. More than one trip can be assigned to an interval. Two conditions must be met. First, the trips can not overlap. Second, the interval should be long enough to cover all trips. These constraints ensure that enough intervals are generated to cover all trips. Equations (2.1b) to (2.1g) in the formulation embody the *task covering constraints*.

Finally, the activities must also have the ability to form combinations. The

[2]If roundtrip durations on all lines are equal then trips do not need to be labelled by their lines and the timetable for many lines can be represented as the timetable for a single line

number of intervals associated with a combination must be a multiple of the total number of intervals on any weekday in a complete cycle of the combination. The distribution of assigned intervals by type should be the same as in the combination. These conditions correspond to the *combination feasibility constraints* (constraints (2.1h) and (2.1i) in the formulation).

2.3.1 Exact formulation

Consider the following sets, parameters and variables.

Sets:

I: the set of daily trips; i refers to an individual trip.

J: set of intervals; j refers to an individual interval. Some of these intervals will not be used. The cardinality of this set is equal to $|I|$.

S: the set of combinations in the combination menu; s refers to an individual combination.

F: the set of interval types; f refers to an individual interval type.

$F(s)$: the set of interval types involved in combination $s, s \in S$; $F(s) \in F$.

Parameters:

t_i: the time when trip i must be started, $i \in I$.

r_i: the roundtrip duration of trip i, $i \in I$.

b_s: the total number of intervals on any weekday of combination $s, \forall s \in S$.

γ_f^s: the number of intervals of interval type f on any weekday of combination $s, f \in F(s), s \in S$. Thus, $\sum_{f \in F(s)} \gamma_f^s = b_s, \forall s \in S$.[3]

l_f: the length of the intervals of interval type f, $f \in F$.

c^s: the wage (expressed in $\$/hr$) paid to a driver working combination s, $s \in S$.

Decision variables:

$$\delta_{ij} = \begin{cases} 1 & \text{if trip } i \text{ is assigned to interval } j, i \in I, j \in J, \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{jf}^s = \begin{cases} 1 & \text{if interval } j \text{ corresponds to type } f \text{ and combination } s, \\ & \qquad j \in J, f \in F(s), s \in S, \\ 0 & \text{otherwise.} \end{cases}$$

z_s = number of interval batches (of size b_s) assigned to combination $s, s \in S$.

Auxiliary variables:

[3]As an illustration consider combination examples in previous tables. In those examples: $b_1 = 1$, $\gamma_8^1 = 1$; $b_2 = 5$, $\gamma_4^2 = 0.2$, $\gamma_9^2 = 0.8$; $b_3 = 7$, $\gamma_{4m}^3 = \frac{2}{7}$, $\gamma_{4e}^3 = \frac{2}{7}$, $\gamma_8^3 = \frac{3}{7}$; $b_4 = 5$, $\gamma_5^4 = 0.4$, $\gamma_{10}^4 = 0.6$; $b_5 = 5$, $\gamma_4^5 = 0.4$, $\gamma_{12}^5 = 0.4$, $\gamma_8^5 = 0.2$; $b_6 = 10$, $\gamma_4^6 = 0.2$, $\gamma_6^6 = 0.2$, $\gamma_{10}^6 = 0.5$, $\gamma_0^6 = 0.1$; $b_7 = 10$, $\gamma_2^7 = 0.1$, $\gamma_4^7 = 0.2$, $\gamma_{10}^7 = 0.6$, $\gamma_0^7 = 0.1$

$l_{(j)} =$ length of interval $j, j \in J$

$$x_j = \begin{cases} 1 & \text{if interval } j \text{ will be used, } j \in J \\ 0 & \text{otherwise.} \end{cases}$$

The problem can be formulated as a mathematical program as follows:

$$\min \sum_{s\in S, f\in F(s)} \sum_{j\in J} c^s l_f x^s_{jf} \tag{2.1a}$$

subject to

$$\sum_{j\in J} \delta_{ij} = 1 \qquad \forall i \in I \tag{2.1b}$$

$$x_j \stackrel{def}{=} \sum_{s\in S, f\in F(s)} x^s_{jf} \leq 1 \qquad \forall j \in J \tag{2.1c}$$

$$\delta_{ij} \leq x_j \qquad \forall i \in I, j \in J \tag{2.1d}$$

$$l_{(j)} \stackrel{def}{=} \sum_{s\in S, f\in F(s)} x^s_{jf} l_f \qquad \forall j \in J \tag{2.1e}$$

$$(\delta_{ij} + \delta_{kj} - 1)(t_i + r_i) \leq t_k \qquad \forall i \in I, j \in J, k \in I, t_i \leq t_k \tag{2.1f}$$

$$(\delta_{ij} + \delta_{kj} - 1)(t_k + r_k) - t_i \leq l_j \qquad \forall i \in I, j \in J, k \in I, t_i \leq t_k \tag{2.1g}$$

$$\sum_{j\in J} \sum_{f\in F(s)} x^s_{jf} = b_s z_s \qquad \forall s \in S \tag{2.1h}$$

$$\sum_{j\in J} x^s_{jf} = \frac{\gamma^s_f}{b_s} \sum_{j\in J} \sum_{f\in F(s)} x^s_{jf} \qquad \forall f \in F(s), \forall s \in S \tag{2.1i}$$

$$\delta_{ij} \in 0, 1 \qquad \forall i \in I, j \in J \tag{2.1j}$$

$$x^s_{jf} \in 0, 1 \qquad \forall j \in J, f \in F(s), s \in S \tag{2.1k}$$

$$z_s \in \aleph \qquad \forall s \in S. \tag{2.1l}$$

The goal (2.1a) is to minimize the cost paid in wages. Constraints (2.1b) ensure that each trip is assigned to an interval while constraints (2.1c) force each chosen interval to be of only one type and combination. Constraints (2.1d) guarantee that trips are not assigned to non-chosen intervals. Constraints (2.1e - 2.1g) impose time conditions. In equations (2.1e) the length of each interval is computed. Constraints (2.1f) prevent overlapping trips within the same interval. Finally, constraints (2.1g) ensures that each interval is long enough to cover its trips [4]. Constraints (2.1h) ensure that intervals are associated to a combination s in batches of size b_s (this implies that drivers are hired in batches of size equal to the combination cycle). Constraints (2.1i) ensure that for each combination the proportion of intervals of each type must coincide with the proportion of intervals of the interval type on any weekday of the combination. If split shifts are part of a combination two extra constraints need to be added. First, the morning and afternoon intervals must be worked around the morning and afternoon peak hours, respectively. Second, the number of intervals of a morning type and its corresponding evening type must coincide. Problem (2.1) is of combinatorial nature and hard to solve.

The above formulation yields the assignment between trips and intervals for one day. The morning and afternoon intervals of each split shift can then be linked to obtain all the shifts to be worked on any weekday.

A feasible solution for this problem must satisfy the *driver-trip availability property*: at any moment the number of active tasks ($v(t)$) can not exceed the number of active intervals ($n(t)$). The number of active tasks is obtained directly from the timetables. The number of active intervals can be obtained indirectly from the

[4] These constraints ensure that an interval can cover all trip pairs (i, k) assigned to it. This is sufficient because $l_{(j)}$ is then large enough to cover the first and last trip, and therefore all the trips in between.

solution of problem (2.1) as is shown next.

To identify active intervals at a certain moment, the intervals must have a clear starting time. However, problem (2.1) does not specify the starting time of each interval. If an interval and its duties are equally long, then the starting time must coincide with the beginning of the first trip. Otherwise, the solution would not be unique since the slack in each interval length can be allocated at the beginning or at the end of the interval (that is, the driver who works it can start or finish on an idle period). Let's assume that intervals begin when their first trip departs. Then, the starting time of an interval ($t_{(j)}$) can be computed as:

$$t_{(j)} = \min\{t_i : \delta_{ij} = 1\} \tag{2.2}$$

Then, the number of active intervals correspond to the sum of started but not finished intervals. This number is computed using the following variable:

$$\theta^t_{fs} = \text{number of intervals of type } f \text{ and combination } s \text{ to start at time } t,$$

$$\forall f \in F(s), \forall s \in S$$

Then θ^t_{fs} can be computed as:

$$\theta^t_{fs} = \sum_{j \in J} x^s_{jf} h\{t_{(j)} = t\} \tag{2.3}$$

Where $h\{\cdot\}$ is a function taking the value 1 if its argument is true and 0 otherwise. Now, the driver-trip availability property can be expressed as:

$$n(t) = \sum_{s\in S, f\in F(s)} \sum_{u=t-l_f}^{t} \theta_{fs}^{u} \geq v(t) \tag{2.4}$$

In the next section a relaxed formulation of problem (2.1) is introduced. This new model is built using constraint (2.4) instead of the task covering constraints.

2.3.2 Simplified formulation

In this section an alternative model to formulation (2.1) is proposed. The optimal solution for this alternative model will most likely be infeasible (as shown later) but will provide a lower bound on the minimum cost. A heuristic to find a feasible solution based on the output of this simplified model is proposed later. The gap between these lower and upper bounds is shown to be tight for planning purposes.

In this approximated formulation time is discretized in small intervals [5]. The number of active tasks, $v(t)$, is discretized accordingly as v_t. Figure 2.6 shows v_t for the timetable of figure 2.4, assuming a roundtrip duration of one hour for all lines.

The relaxed set covering problem replaces (2.1b-2.1g) by (2.4) and does so in discrete time. The goal is to find a set of intervals that cover the curve ignoring the trip-interval pairing. Let's define n_t as the number of active intervals at time t.

[5] Notice that vehicles are already scheduled to depart at discrete instants, defining a natural discretization of time for interval starting times, say five minutes.

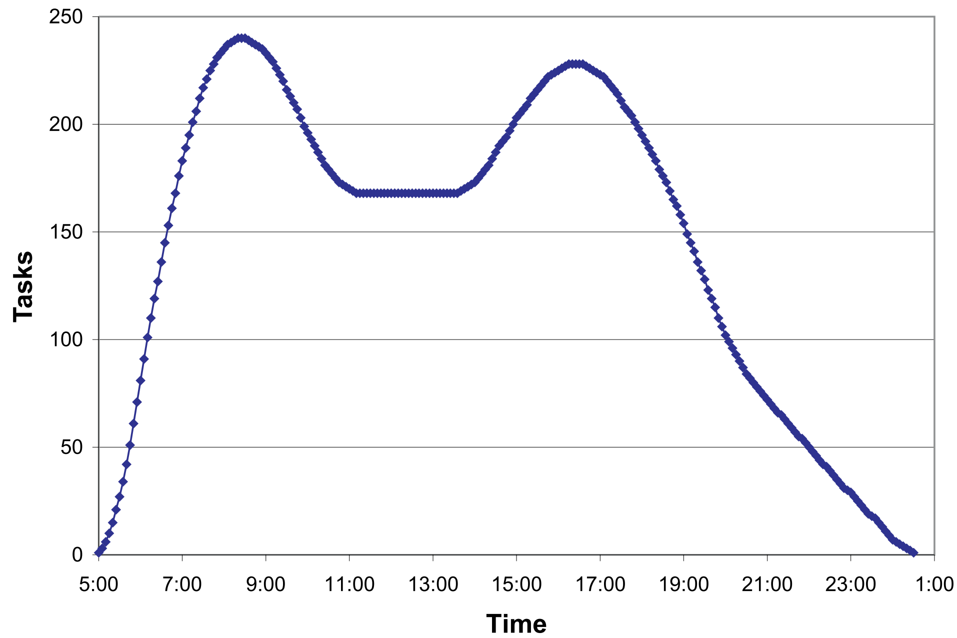

Figure 2.6: Number of vehicles running at a given time according with the timetable of figure 2.4

The following auxiliary variables are useful:

$\theta_{fs} = \sum_{t=1}^{T} \theta_{fs}^{t}$: Number of intervals of interval type f, $f \in F(s), s \in S$.

$\theta_s = \sum_{f \in F(s)} \theta_{fs}$: Number of intervals associated with combination $s, s \in S$.

The relaxed model would be:

$$min \sum_{s \in S} 40 c^s \theta_s \tag{2.5a}$$

subject to

$$n_t \stackrel{def}{=} \sum_{s\in S, f\in F(s)} \sum_{u=t-l_f}^{t} \theta_{fs}^u \geq v_t \qquad \forall t \in [1,T] \tag{2.5b}$$

$$\theta_s \stackrel{def}{=} \sum_{f\in F(s)} \theta_{fs} = b_s z_s \qquad \forall s \in S \tag{2.5c}$$

$$\theta_{fs} \stackrel{def}{=} \sum_{t=1}^{T} \theta_{fs}^t = \frac{\gamma_f^s}{b_s} \sum_{t=1}^{T} \sum_{f\in F(s)} \theta_{fs}^t = \frac{\gamma_f^s}{b_s}\theta_s \qquad \forall f \in F(s), \forall s \in S \tag{2.5d}$$

$$\theta_{fs}^t \in \aleph \quad \forall f \in F(s), \forall s \in S, \forall t \in [1,T] \tag{2.5e}$$

$$z_s \in \aleph \qquad \forall s \in S. \tag{2.5f}$$

Here, (2.5b) replaces the time task covering constraints while the combination feasibility constraints are preserved. As in the exact model, extra constraints must be added if split shifts are included.

This integer programming model is considerably more tractable than problem (2.1). Indeed it is shown below that the coefficient matrix of the left hand side of the constraints (2.5b) is totally unimodular. Therefore, since v_t is an integer vector, all vertices of the polyhedron defined by those constraints are integer vectors. Thus, in the absence of constraints (2.5c) and (2.5d) constraints (2.5e) and (2.5f) could be dropped. Fortunately, the number of constraints of type (2.5c) and (2.5d) is small:

only $\sum_{s\in S}(|F(s)|-1)$ constraints of type (2.5c) while less than $|S|$ of type (2.5d).[6] Therefore the solution to (2.5) is readily obtained by solving its LP relaxation.

The structure of the matrix of the left hand side of constraints (2.5b) is shown in figure 2.7. The matrix is composed of as many squared submatrices as shift types in all combinations in the combination menu. The only non-zero elements in each squared submatrix are all the elements of a thick diagonal fringe taking the value "1". The thickness of the diagonal depends of the length of intervals in each interval type (l_f). To prove that the matrix is totally unimodular consider the following procedure. Starting from the top row and moving sequentially to rows below, redefine each row by subtracting from it the sum of all rows above it. The resultant matrix has two kinds of columns: those with one non-zero element equal to 1 and those with two non-zero elements, a 1 and a -1. It is well known that a

[6]Often, a solution satisfying constraints (2.5d) also satisfies constraints (2.5c). Constraints (2.5c) are only needed if the minimum common multiple among $\gamma_f^s, \forall f \in F(s)$ is not 1. Since then both sides of

$$b_s\theta_{fs} = \gamma_f^s \sum_{f\in F(s)} \theta_{fs}, \forall f \in F(s), \forall s \in S$$

could be divided by the minimum common multiple and a new equivalent equation with a lower b_s would be obtained indicating that the equation allows intervals associated with combination s to be picked in batches smaller than b_s. However, if the minimum common multiple among $\gamma_f^s, \forall f \in F(s)$ is 1, then for both sides of (2.5d) to be integer, $\sum_{f\in F(s)}\theta_{fs}$ must be a multiple of b_s. Indeed, these constraints are not needed for any of the combinations 1-7 presented before.

matrix of this characteristics is totally unimodular (see [24]).

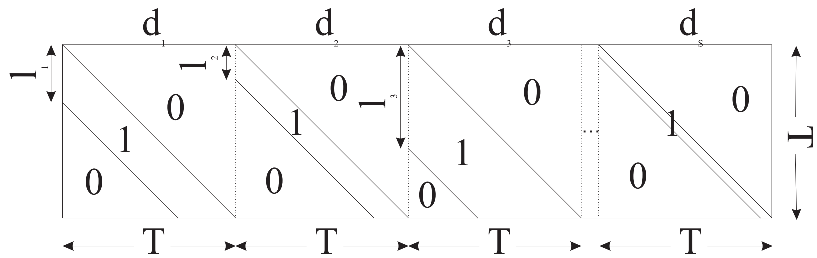

Figure 2.7: Coefficient matrix of problem (2.5)

Problem (2.5) is not valid for the shift filling problem since it does not restrict (fractions of) trips to be assigned to the same interval on consecutive time periods. Thus, it assumes that drivers (not the vehicles) can travel infinitely fast. This is now illustrated with two examples.

As a first example consider figure (2.8a) which shows a timetable given by three two-periods trips to be covered by intervals. The trips must be started at $t = 1, t = 2$ and $t = 3$ yielding the function v_t of the figure. Consider the case where all intervals are three periods long. In this case, the approximated model would predict that two intervals are enough to cover these trips. One interval from $t = 1$ to $t = 4$, while the other from $t = 2$ to $t = 5$. In this solution one of the trips (e.g. the trip starting at $t = 3$) would be assigned to both intervals, one half of the trip to each. However this solution is not feasible for the shift filling problem since the driver working the second interval would end a trip at $t = 4$ in the garage and should immediately finish the incomplete task left by the driver working the other interval. This is infeasible since this last task can not be started at the garage.

In the above example, the shift lengths and the roundtrip durations are of a similar order of magnitude. However the focus of this dissertation is on systems where the length of a shift is considerably longer than the roundtrip duration. Unfortunately, in this case it is also easy to build an example where the solution of model (2.5) is not feasible for the shift filling problem. Consider figure 2.8b which shows five two-periods trips to be completed by six time period intervals. The solution of problem (2.5) would predict again that only two intervals are needed to cover these trips. However, for the same reasons explained before, this solution is not feasible for the real problem. Notice that this occurs even though the length of the shifts was a multiple of the length of the trips. In both cases the real minimum cost exceeds the simplified model's prediction by 50%. Fortunately, timetables usually have a very different shape (double-peaked as argued before) from those of these examples and it is shown in the next two sections that for real transit data the simplified model predicts the costs very well. Additionally, experience shows that the bounds improve when the size of the shift filling problem grows. This is very valuable since problem (2.1) becomes less tractable with increasing size.

In summary, the approximated formulation provides a lower bound on the driver cost for the deterministic shift filling problem. Notice that the more alternatives in the combination menu, the lower this lower bound will be. However, this lower bound can not drop beyond the area under v_t multiplied by the lowest salary paid to a combination in the menu.

Figure 2.8: Simple examples to show that infeasible solutions are allowed by model (2.5); a) Case when shift lengths and roundtrip durations are of similar order, b) case when shift lengths are considerably longer than roundtrip durations

Since problem (2.5) is shown to be accurate, the impact of considering more heterogeneous combinations is estimated by solving problem (2.5) for different combination menus. It is shown in the next section that adding this flexibility can lead to lower possible costs.

2.4 The Deterministic Problem: Interpretation and validation

In this section problem (2.5) is solved for a variety of combination menus. In each case, the solution is later used to identify a feasible solution for problem (2.1). In all cases studied here the gap between the costs of these two solutions is under 1%. Thus, the optimal cost of problem (2.5) is considered a good (under)estimator of the total shift filling cost.

2.4.1 Characteristics of the simplified solution

In this section the impact of the timetable and the combination menu on the optimal cost for the simplified model is explored (thus, all over this section solutions refer to problem (2.5). The timetable was derived based on the AC Transit data and its corresponding task curve is shown on figure 2.6. To identify the impact of the timetable in the solutions, different noon demands are considered. First, single-week combinations are addressed. To illustrate the impact of shorter interval types, combinations breaking the 8-hour per day rule are considered. Later in the section, heterogeneous combinations are studied.

Single shift family case

In this section the simplest combination menu is analyzed: all drivers work a unique combination and this combination is composed of a single interval type. Every day all drivers work for the same number of hours[7].

Assuming a constant hourly wage, it should be expected that the longer the shifts the higher the cost since longer shifts increase the difficulty of finding a close match between drivers and active tasks. In general, this turns to be true. Figure 2.9 shows the active task curve (v_t) according with the timetable and the optimal number of

[7] If this number is different than eight, the drivers will not be working forty hours a week. The purpose of this section is only to explore the effects of shifts of different lengths.

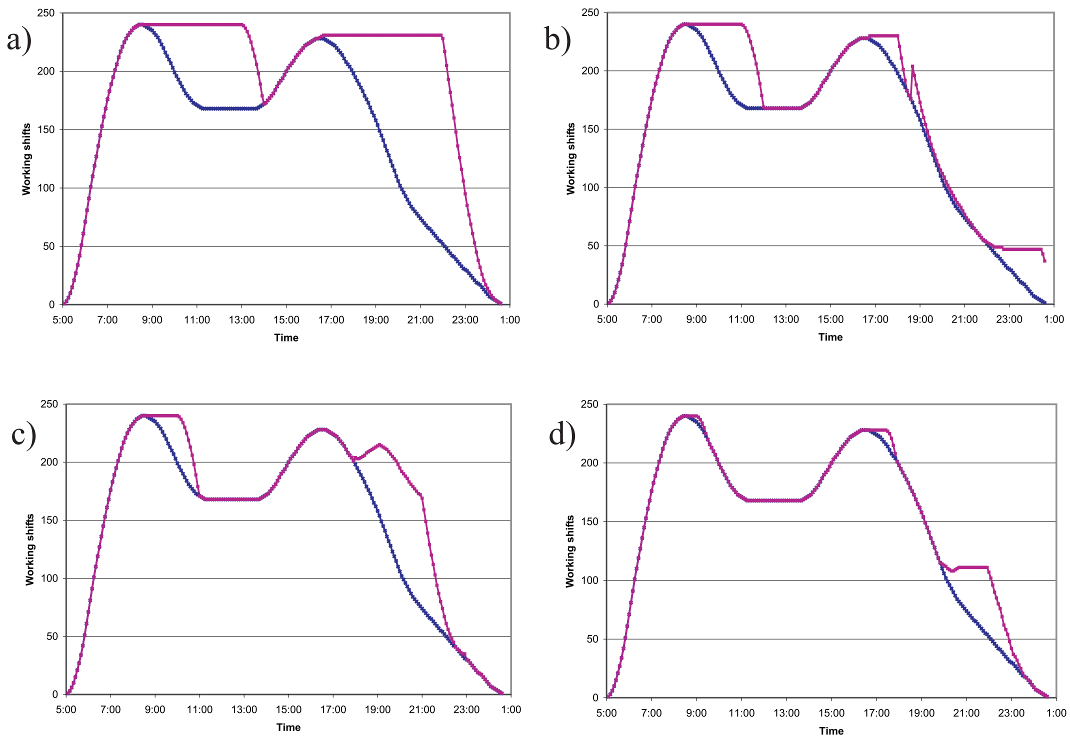

Figure 2.9: Optimal intervals (n_t), and tasks (v_t) when all drivers work shifts of the same length a) n_t (8-hour intervals), b) n_t (6-hour intervals), c) n_t (5 hr intervals), d) n_t (4-hour intervals).

active intervals (n_t) throughout the day if all intervals were to last for either 8, 6, 5 or 4 hours[8].

As expected, considerably fewer idle hours (when the number of active intervals exceed the active tasks) are scheduled for shorter shifts. Figure 2.10a plots the

[8]The optimal solution is almost always not unique complicating the comparison between solutions of different problems. Thus, to facilitate this comparison the optimal solution has been further constrained so that no driver will be idle before the morning peak and no shift should remain active after the last bus is dispatched. None of these constraints affect the optimal cost.

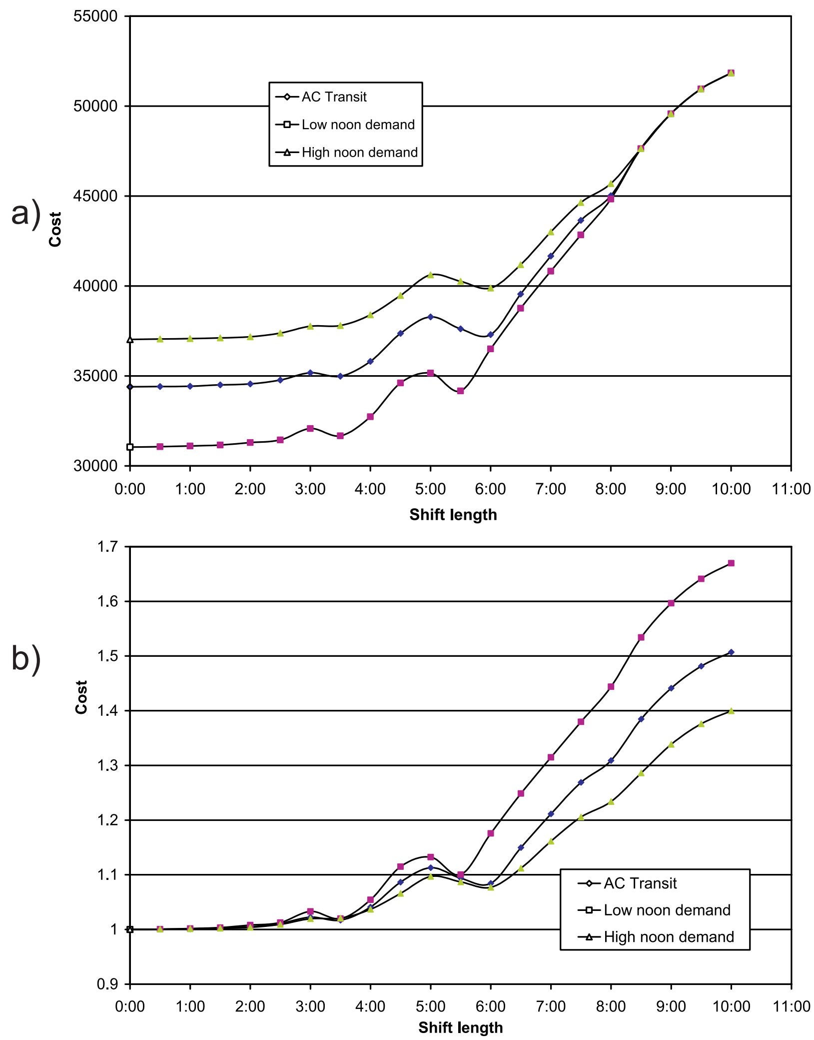

Figure 2.10: Driver cost for continuous shifts; a) Total driver cost as a function of the shift length and the noon demand ratio; b) penalty paid for shifts not being divisible (expressed as a fraction of the totally divisible case).

minimum cost (assuming a wage of $12 per hour of contract work) for shift lengths ranging from 5 minutes to 10 hours. Three active task curves are analyzed: the one based on AC Transit demand, and similar profiles but with higher and lower peak to noon demand ratios. Notice that the peak to noon demand ratio impacts the total cost only for short shifts (in comparison to the typical work-day length). Figure 2.10b presents the same information as figure 2.10a but cost is divided by the minimum cost attainable (e.g. if all intervals were 5 minutes long). The figure shows that operating only with 8 hr intervals imposes a 25% to 40% extra cost to the system, or equivalently that 20% to 30% of the driver-hours schedules are not used.

Notice that although, in general, the longer the interval length the higher the cost, this is not exactly true. For the tasks curve in this example, working with only 5 hour intervals turns to be more expensive than with only 5 1/2 hours and sometimes even than with only 6 hour intervals. This happens because in the solutions, intervals are scheduled in groups. For instance until the morning peak, intervals are continuously scheduled. Five hours later those intervals end and new intervals must be scheduled. In the case of the five-hour interval, this second group of intervals end when they are most needed (evening peak). Therefore, the problem schedules a third set of intervals although some of these drivers will be idle for most of their shifts. The excess of intervals scheduled after the afternoon peak hour in figure 2.9b shows these idle hours. The second group of six-hour intervals end in good synchronization with the drop in departure rate after the evening peak. Notice the right hand side of the spike in the right side of figure 2.9b indicating the end of those shifts.

Indeed, the optimal cost when the agency uses a single type of interval is easy to estimate. It can be assumed that before the morning peak, intervals should be scheduled to start just in time for their first trip. Analogously, after the evening peak, intervals should be scheduled to end simultaneously with their last trip. If these intervals cover the whole active task curve, an optimal solution is obtained. If they do not, the problem should be repeated but only with those trips not covered by the intervals already scheduled.

Impact of split shifts

Unions and agencies usually agree on a guaranteed 8 hours working day for drivers. However, as was discussed in the previous section, if all drivers are hired for eight continuous hours the number of idle hours may exceed 30%. To avoid this cost, agencies often hire drivers to work split shifts at a premium wage due to the perceived inconvenience of these shifts. In a split shift, the driver works some hours (usually four) during the morning peak and some hours (four) in the evening peak. The premium wage is always lower than twice the wage for eight hour continuous shifts (usually is 50% higher); otherwise two continuous eight-hour drivers would be preferred to a split shift.

To estimate the impact of split shifts, a menu of the following two single-week cycle combinations was considered: a) some drivers work an 8-hour interval every day, and b) other drivers work a 4-hour morning and a 4-hour afternoon interval every day. Both combinations were equally paid. The total cost of this menu for the timetable analyzed before reduces the number of idle hours dramatically (more

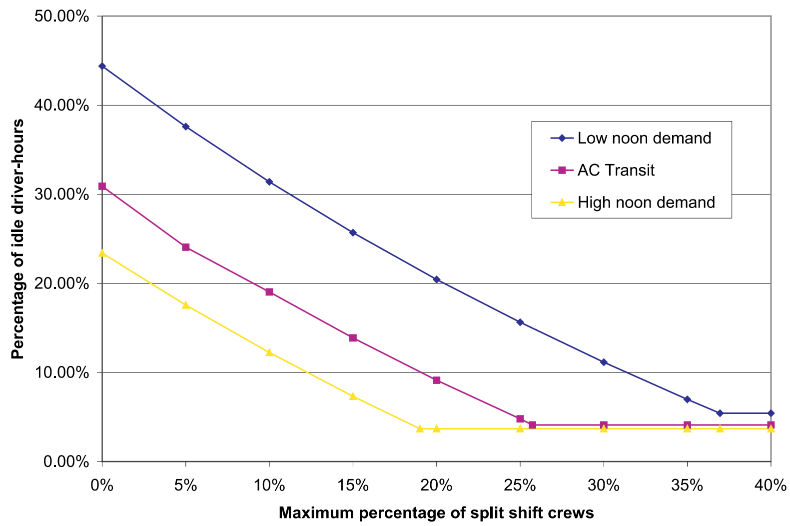

Figure 2.11: Percentage of idle hours for a given maximum fraction of drivers working split shifts

than 85%) compared to a menu of continuous shifts only.

Since drivers dislike split shifts, unions often request that split shifts should not exceed a certain fraction of the workforce. Therefore, the agency should understand the impact of this regulation on idle hours. This relationship is depicted in figure 2.11. The figure shows that the number of idle hours decreases with the maximum fraction of split shifts until a certain threshold is reached. Although it can not be observed in the figure, the marginal benefit of an extra split shift (before the threshold) is equal in all cases to the salary of a driver working 8 continuous hours. Thus by shifting a driver from a continuous eight hour shift to a split shift, the agency saves the wage of a full 8-hour continuous driver.

Impact of trippers

Excessive idle hours can also be avoided by hiring special drivers called trippers to do short tasks. These drivers are usually hired to work peak hour trips and are paid a premium wage. Figure 2.12a shows the optimal number of trips assigned to trippers versus tripper wage when the combination menu is composed of the following two single-week cycle combinations: a) some drivers work an 8-hour interval every day, and b) other drivers work a 1-hour interval every day. Three demand scenarios are analyzed. As expected fewer trippers are hired when their wage increases. Figure (2.12b) shows the total cost versus trippers wage for the same cases.

As with split shifts, the total cost decreases when more trippers are hired until a threshold is reached. Figure 2.13 shows this relation between total cost and maximum number of trippers hired when the wage for trippers doubled the wage for 8-hour continuous shifts. Unlike in the split shift case, the figure indicates that the marginal benefit of a tripper decreases in a piecewise linear manner with the number of trippers already hired. This should be intuitive since before adding any trippers some continuous shifts can be replaced with just one tripper. These continuous shifts will be the first being replaced and the marginal benefit of each replacement will be constant. Once all these shifts have been replaced, the continuous shifts with only two trips assigned should be replaced. The marginal benefit of each replacement should still be constant, but lower. This procedure should continue until the marginal benefit of a replacement is non-positive, in this example, when all

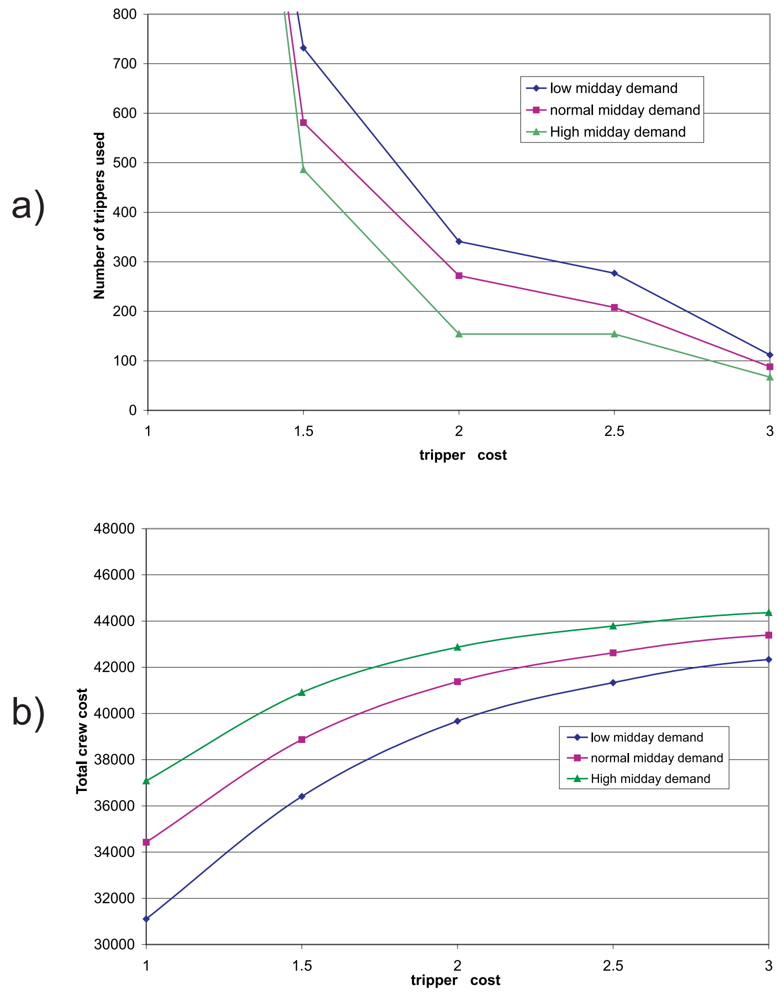

Figure 2.12: Optimal solution when the agency chooses to work with eight hour shifts and trippers for different trippers wage; a) number of trippers used; b) Optimal cost.

continuous shifts are assigned to four or more trips.

The previous sections have analyzed the impact of combination menus already in use by agencies today. In the next section more flexible combinations are explored.

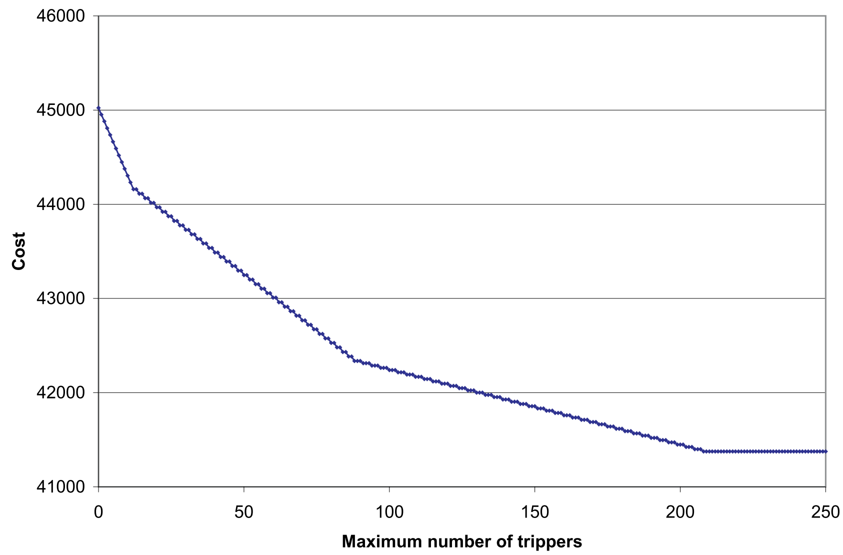

Figure 2.13: Total cost for a maximum number of tripper runs

Heterogeneous combinations

To avoid the idle hours resulting from hiring all drivers for eight continuous hours, agencies offer split shifts, part time shifts, overtime and trippers work. These labor contracts are more expensive than the 8 hour per day contract since they are either resisted by drivers or by unions. However, a more flexible strategy of rotating shifts as explained in section 2.2.2 has not been explored yet. This section shows the benefits of relaxing the 8 hours a day constraint for a 40 hours and 2 days off a week instead.

Consider the following combination menu:

Combination 1 of table 2.1: Eight-hours a day, five-days a week.

Combination 4 of table 2.2: Three days working ten hours and two days only five.

Combination 8: One split shift of four hours in the morning and four hours in the evening five days a week.

Figure 2.14 plots the optimal solution for this menu in which the three combinations are paid the same wage of 12 per hour. As can be observed, the fit is almost perfect: the total cost is just $\$34,512$ or 0.34% over the lower bound (attained if shifts were totally divisible). By having added the heterogenous combination (the second one) into the shift menu, the idle driver hours are reduced by more than 90% (this shift menu without combination 4 was analyzed earlier in this section). Additionally, the new solution uses fewer split shifts (20%). Notice that a new menu can be designed with the same optimal cost in which all combinations include a split shift only once a week.

This last example assumes, although unrealistically, that split shifts are paid at the same wage that other shifts. This was done to isolate the impact of the new combination. If split shifts are better paid than continuous shifts the effect of the heterogeneous combination should also be significant since then fewer split-shift combinations and more of the new combinations would be used. However, timetable and drivers can be closely matched using heterogeneous combinations even if split shifts are not considered as is now shown.

Consider the following six combinations already explained in section 2.2.2:

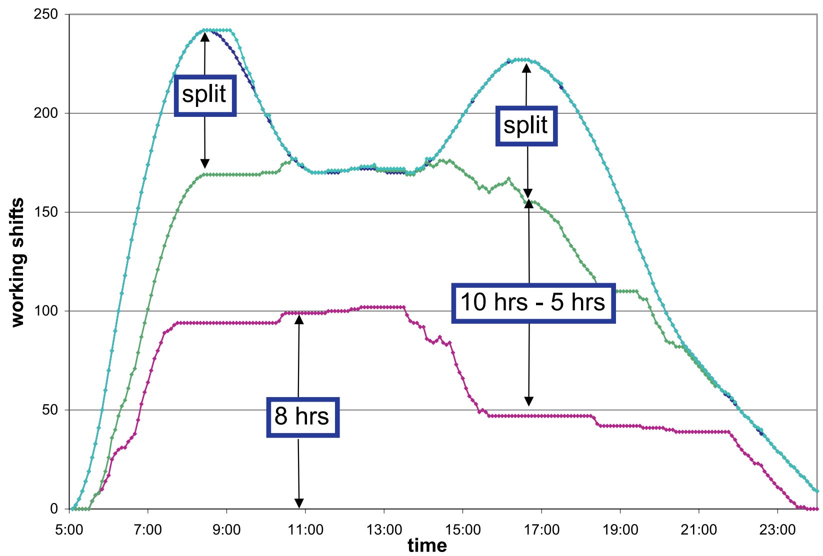

Figure 2.14: Optimal solution when the agency chooses to work with eight hours, split and combination (10,5) shifts

Combination 1. 8 hours a day, 5-days a week

Combination 2. four days working nine hours and one day working only four

Combination 4. three days working ten hours and two days working only five

Combination 5. two days working twelve hours, one day working eight hours and two days working four hours a week.

Combination 6. First week: ten hours four days a week (no work on weekends); second week: four hours two days, six hours two days and ten hours on a weekday and on Saturday,

Combination 7. First week: ten hours four days a week (no work on weekends); second week: two hours one day, four hours two days and ten hours two days and on Saturday.

If the agency operates with combination 1 only, then the optimal cost for problem (2.5) is $\$45,216$. If instead combinations 1, 2 and 4 are used, the total cost drops to $\$41,472$. If combinations 6 and 7 are added to the menu then the optimal cost drops to $\$40,320$. Finally, if combination 5 is added too, the cost drops to $\$34,416$ the same cost as if all shifts were to work for blocks of one hour and still receive the same wage.

This last solution requires 6 drivers working combination 1, 25 drivers working combination 2, 15 drivers working combination 4, 190 drivers working combination 5, 120 drivers working combination 6 and 30 drivers working combination 7. As explained before, since the match between the number of active tasks and active intervals is perfect throughout the day, and all shift lengths are a multiple of the roundtrip duration, the solution is also feasible (and optimal) for the exact problem (2.1). This shows the substantial benefits that agencies can achieve by relaxing the eight hour a day 5 days a week agreement. Not only they can have the desired number of drivers at each moment but they can do it without expensive and inconvenient split shifts.

2.4.2 Approximated solution for the exact problem

The intervals of the simplified formulation (2.5) would cover all tasks if these were infinitely divisible. Since this is not the case (2.5) does not necessarily provide a feasible solution for the shift filling problem. Instead, it provides a lower bound. To obtain an upper bound, a simple heuristic is developed where a feasible solution is identified based on the solution of the simplified model (2.5). This heuristic is described in detail in Appendix A. However, an upper bound can be obtained analytically too as is shown next.

Any solution to problem (2.5) where all trips can be assigned undivided to an interval is also feasible for problem (2.1). However, a solution to problem (2.5) might not have this property. If all intervals reserve some slack equal in length to the longest trip, R, then the solution would be feasible for problem (2.1). For example, if all intervals in the menu are of length S then a feasible solution is obtained by solving (2.5) with intervals of length $S - R$ assuming that wages are $\frac{S}{S-R}$ times higher. For instance, if $S = 8$ hours and $R = 1$ hour then an upper bound is obtained by solving the problem for $S = 7$ hours and multiplying the cost by $\frac{8}{7}$. For the example of figure 2.10(a) the cost for $S = 7$ hours is $\$41,664$. Then $\frac{8}{7} * \$41,664 = \$47,616$ is an upper bound for the problem. The lower bound (obtained from the same figure for $S = 8$ hours) is $\$45,024$, yielding a gap of 5.75%.

This procedure ensures that the last trip assigned to an interval never has to be divided, regardless of the assignment. Since the indivisibility condition only has to be satisfied for one assignment there is much room for improvement. The heuristic described in Appendix A does not assign any slack. Instead, it assigns the trips in the timetable to the intervals in the simplified model using a greedy logic. If a feasible solution for the shift filling problem is not found, some trips remain unassigned. The heuristic covers those trips by hiring extra drivers.

Examples presented in the next three sections and later in this dissertation show that the gap between the lower and upper bounds obtained with this greedy heuristic is usually under 1%.

Only eight-hour drivers

Consider the case when the combination menu is made only of combination 1 of table 2.1, that is eight hours a day, five-days a week. Since the two peaks are separated by more than eight hours, their active tasks must be covered by different intervals. Hence, a lower bound for the number of intervals is the sum of the active tasks on the two daily peaks. However, once those intervals are scheduled, off-peak periods could still remain understaffed. Then, extra intervals must be added.

As an example look at figure 2.15a displaying a piecewise linear active task curve. There, the horizontal lines represent intervals. These intervals have been assigned to cover the peaks with as much overlap in the mid-day as possible. Figure 2.15b shows the number of active intervals as a function of time for this allocation. Since no extra intervals have to be added, the allocation is optimal for problem (2.5).

Consider now $v(t)$ from figure 2.6, its underlying trips and the same combination menu. In this case, the heuristic in Appendix A found a feasible trip-interval assignment based on the intervals of problem (2.5). Therefore the upper bound and lower bound coincided.

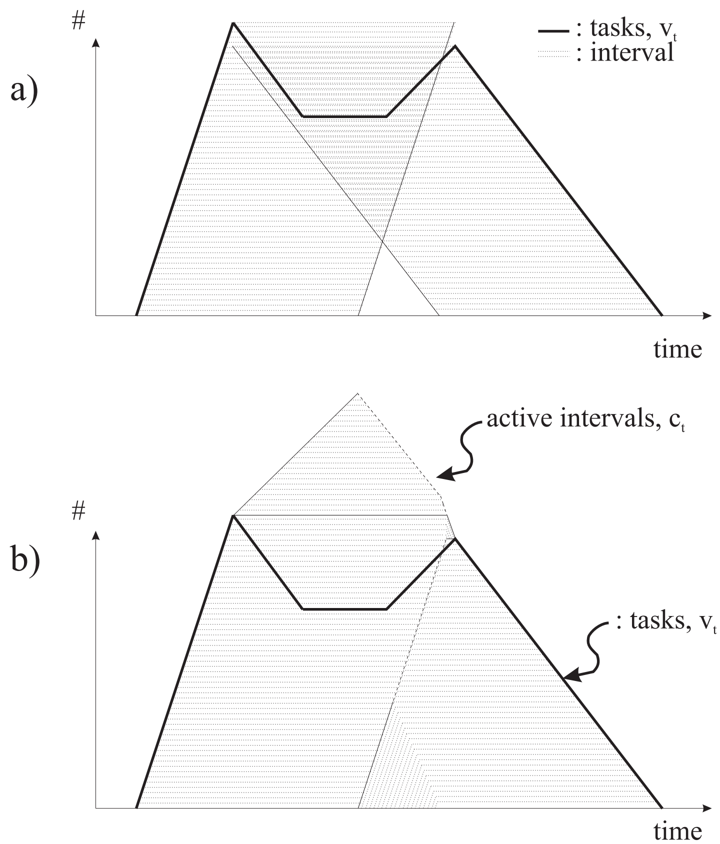

Figure 2.15: Optimal solution for a piecewise linear task curve when the agency can only work with eight hour shifts; a) Intervals; b) Active intervals

Eight-hour drivers and trippers

Consider a combination menu composed of the following two single-week cycle combinations: a) some drivers work an 8-hour interval every day, and b) other drivers work a 1-hour interval every day (trippers). Drivers working the second combination are paid twice as much the wage for those working the first combination.

As illustrated in figure 2.13, an eight continuous hour shift that will be idle more than four hours shouldn't be scheduled since then hiring four trippers instead would be more convenient. Based on this analysis, figures 2.16a and b display two optimal solutions for problem (2.5) for the piecewise linear case analyzed in the previous section.

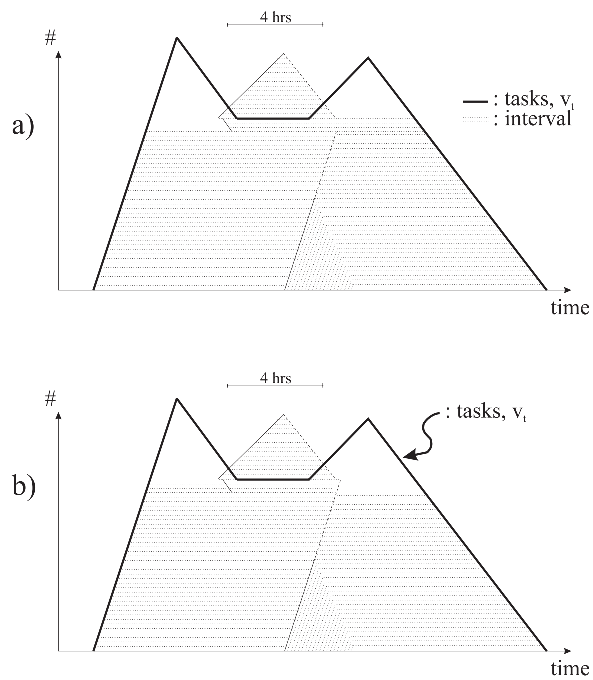

Figure 2.16: Optimal solution to problem (2.5) for a piecewise linear demand example when the agency can work with eight hour shifts and trippers at twice the wage rate; a) Optimal solution; b) Alternative optimal solution

Consider now $v(t)$ from figure 2.6, its underlying trips and the same combination menu. In this case, problem (2.5) yields a total cost of $\$41,496$. Starting from this solution, the heuristic of Appendix A yielded a feasible solution for problem (2.1) with a cost of $\$41,640$. Therefore, the gap was only 0.35%.

Eight-hour drivers, combination of ten and five hours, split shifts and trippers

If a more complete combination menu is considered, for instance:

Combination 1 of table 2.1: Eight hours a day, five-days a week paid at \$12 per hour.

Combination 4 of table 2.2: Three days working ten hours and two days only five paid at \$12 per hour.

Combination 8: One split shift of four hours in the morning and four hours in the evening five days a week paid at \$18 per hour.

Combination 9: One hour a day, five days a week paid at \$24 per hour.

and the set of trips corresponds to the underlying ones on figure 2.6, then the approximated problem (2.5) provides a lower bound of $\$36,984$. Using the greedy heuristic of Appendix A, this solution can be made feasible for problem (2.1) by adding trippers when needed. In this case the cost grows to $\$37,608$, that is only 1.69% more. However after a 2-opt procedure in which couples of trips are swapped between intervals if the total cost drops, a better upper bound is reached: $\$37,152$ which yields a gap of only 0.45%. Finally, if the agency is allowed to pay overtime to the regular drivers and therefore extend their workload at the marginal wage of the trippers, then the cost drops further to $\$37,088$ yielding a gap of only 0.28%.

Thus, for all examples analyzed in this research, the lower bound obtained with the simplified problem (2.5) is within 0.5% of the optimal cost. Notice that if drivers were allowed to start or end their shifts at relief points other than the hub, the upper bound cost would be even closer to the lower bound. Notice too that the richer the combination menu, the closer the gap.

In the next section the impact of absent drivers is explored.

2.5 The Stochastic Problem: Absenteeism

This section considers the shift filling problem with driver absenteeism. On any given day the agency might find itself understaffed at some moments. Therefore, before designing the system the agency must determine a policy to respond under each possible scenario. This daily problem is called in this dissertation the control mechanism. In this case the agency has at least three options: fail to provide the promised schedule, pay overtime, or hire trippers to do the extra work. In this work the first alternative is not be considered. Recall that all drivers can work any line.

The control mechanism works as follows: at the beginning of each day the agency knows exactly which drivers will show up and therefore the times at which each of them start and end working (as stated in their contracts). Then, the agency will try to maximize the number of trips assigned to those drivers (including fractions of trips). To do this an extension of the heuristic in Appendix A will be used. The agency will first use overtime to finish partially assigned trips and trippers for the rest. Trippers and overtime are paid at the same premium wage.

The questions in this case are very similar to the deterministic case: (1) how many drivers should the agency hire for each combination?, and (2) which shift sequence should each of them work to minimize the *expected* cost over the design period? Notice that the contracts will not be changed during that time horizon, and therefore the agency might find itself overstaffed in the morning and understaffed in the evening and still will not be able to shift morning drivers to the evening.

Since trippers and overtime are more expensive than regular drivers, the agency should hire more regular drivers than would be needed to handle an average day.

As in previous sections, some assumptions must be added to make this problem simpler. First, the no-show rate, $(1-p)$, will not exceed 20%, and, second, drivers are independent and identical, showing up each day with probability p.

Section 2.5.1 and 2.5.2 analyze the case where all combinations are equally paid. Section 2.5.1 studies the time independent problem. The results from this section are then used in section 2.5.2 for the time dependent problem. Finally, section 2.5.3 extends these results for the case with heterogeneous wages.

2.5.1 The time independent problem: homogeneous wages

Consider a problem where $v(t) = v$, independent of time. Drivers should be hired so that the number of drivers scheduled to work at any given time, $n(t) = n$, is time independent and satisfies $n \geq v$. Regular drivers are paid w per unit of time if they come to work (regardless of whether or not they work) while trippers (if needed) are paid $w_t (w_t > w)$ per unit of time.

Let q be the number of drivers who show up for a time interval $[t, t + dt]$, $q \leq n$. Then q has a binomial distribution with parameters n and p, and the expected cost per unit time for the interval is:

$$\begin{aligned} E[\text{cost per unit time}] &= wE[q] + w_t \sum_{q=0}^{v-1} \binom{n}{q} p^q (1-p)^{n-q} (v - q) \\ &= wnp + w_t \sum_{q=0}^{v-1} B(q; n, p)(v - q) \end{aligned}$$

Since the problem is time independent, this is true for all time intervals. For this application v and n are large enough numbers, allowing the binomial distribution to be approximated by a continuous normal density with mean np and variance $np(1 - p)$:

$$B(x; n, p) \approx \phi(x; np, np(1 - p))$$

This approximation is reasonable if the expected value is two standard deviations above zero [9]. That is:

$$np - 2\sqrt{np(1 - p)} \geq 0 \tag{2.6}$$

[9]This implies that the probability of a negative q is only 0.02

since it was assumed that $p \geq 0.8$, then inequality (2.6) is satisfied as long as:

$$n \geq 1$$

Then, the expected cost can be expressed as:

$$\begin{aligned}
&E[\text{cost per unit time}] = \\
&\quad = wnp + w_t \int_{x=-\infty}^{v} \phi(x; np, np(1-p))(v-x)dx \\
&\quad = wnp + w_t \left[v\Phi(x; np, np(1-p)) - \int_{x=-\infty}^{v} x\phi(x; np, np(1-p))dx \right] \quad (2.7)
\end{aligned}$$

Where $\Phi(x; \mu, \sigma^2)$ represents the cumulative probability for a normal distribution function with mean np and variance $np(1-p)$ evaluated at x. To minimize this function, the following properties of a normal distribution are used:

- $\frac{\partial \Phi(x;\mu,\sigma^2)}{\partial x} = \phi(x; \mu, \sigma^2)$
- $\frac{\partial \phi(x;\mu,\sigma^2)}{\partial x} = -\frac{(x-\mu)}{\sigma^2}\phi(x; \mu, \sigma^2)$
- $\Psi(x; \mu, \sigma^2) \stackrel{def}{=} \int_{-\infty}^{x} \Phi(x; \mu, \sigma^2)dx = \sigma^2\phi(x; \mu, \sigma^2) + (x-\mu)\Phi(x; \mu, \sigma^2)$

The last property is just a definition of the function Ψ. This function is convex in x. Thus, the total cost per unit time can be expressed as:

$E[\text{cost per unit time}] =$

$$= wnp + w_t\left[(v-np)\Phi(v;np,np(1-p)) + np(1-p)\phi(v;np,np(1-p)))\right]$$

$$wnp + w_t\Psi(v;np,np(1-p))$$

Therefore, the optimal value n^* must satisfy:

$$\frac{d\Psi(v;np,np(1-p))}{dn} = -\frac{wp}{w_t} \tag{2.8}$$

The following analysis [10] is done to estimate the left hand side of 2.8:

$$\frac{d\phi(x;np,np(1-p))}{d(np)} = -\frac{\partial\phi(x;np,np(1-p))}{\partial x} + \frac{\partial\phi(x;np,np(1-p))}{\partial(np(1-p))}(1-p) \tag{2.9}$$

Since p is assumed close to 1, and the normal probability density is more sensitive to the mean than to the variance, the second term of this expression is neglected. Thus,

$$\frac{d\phi(x;np,np(1-p))}{d(np)} \simeq -\frac{\partial\phi(x;np,np(1-p))}{\partial x} \tag{2.10}$$

Therefore, the left hand side of (2.8) can be approximated as:

[10]Using $\frac{d\phi(x;\mu,\sigma^2)}{d\mu} = -\frac{\partial\phi(x;\mu,\sigma^2)}{\partial x}$

$$\int_{-\infty}^{v}\int_{-\infty}^{y}\frac{d\phi(x;np,np(1-p))}{dn}dxdy \simeq \int_{-\infty}^{v}\int_{-\infty}^{y}\left(-p\frac{\partial\phi(x;np,np(1-p))}{\partial x}\right)dxdy = $$
$$= -p\Phi(v;np,np(1-p))$$

Thus, the optimal number of drivers to hire, n^*, can be obtained by solving the following:

$$\Phi(v;np,np(1-p)) \simeq \frac{w}{w_t} \tag{2.11}$$

Since n^* tends to v/p as v grows, a short formula can be obtained by approximating the left hand side of equation (2.11) as:

$$\Phi(v;n^*p,n^*p(1-p)) \simeq \Phi(v/p;n^*,v(1-p)) = 1-\Phi(n^*;v/p,v(1-p)) \tag{2.12}$$

and therefore getting:

$$n^* \simeq \bar{n} = \Phi^{-1}\left(1-\frac{w}{w_t};v/p,v(1-p)\right) \tag{2.13}$$

where $\Phi^{-1}(x;\mu,\sigma^2)$ represents the inverse of the cumulative normal distribution with mean μ and variance σ^2 for a cumulative probability x. This relation is illustrated in figure 2.17. Notice that $\bar{n}$ might not be integer.

The quality of this approximation is shown in figures 2.18a and b. These figures display $(\frac{n^*-v}{v})$ and $(\frac{\bar{n}-v}{v})$ versus the show-up rate p for different number of active

Figure 2.17: Determining n^*, given v

tasks (v). Figure 2.18a considers the case of a low ($w_t = 1.5w$) tripper wage while figure 2.18b the case of a very high one ($w_t = 4w$). As expected, the larger the number of active tasks the better the approximation. Similarly, the lower the absenteeism rates the better the approximation. Figures 2.18a and b also include the function $1/p - 1$, representing the extra drivers that would have to be hired so that the expected number showing up would match the number of active tasks. Notice from equation (2.11) that if $\frac{w}{w_t} \geq 0.5$ or equivalently if $w_t \leq 2w$ then $n^* \leq v/p$ and if $w_t \geq 2w$ then $n^* \geq v/p$. This is reasonable because the agency should avoid overstaffing with inexpensive trippers. In the limit, if $w = w_t$, the agency should hire just trippers. This is consistent with figures 2.18a and b where the overstaffing curves lay over $1/p - 1$ when ($w_t = 4w$) and under it when ($w_t = 1.5w$).

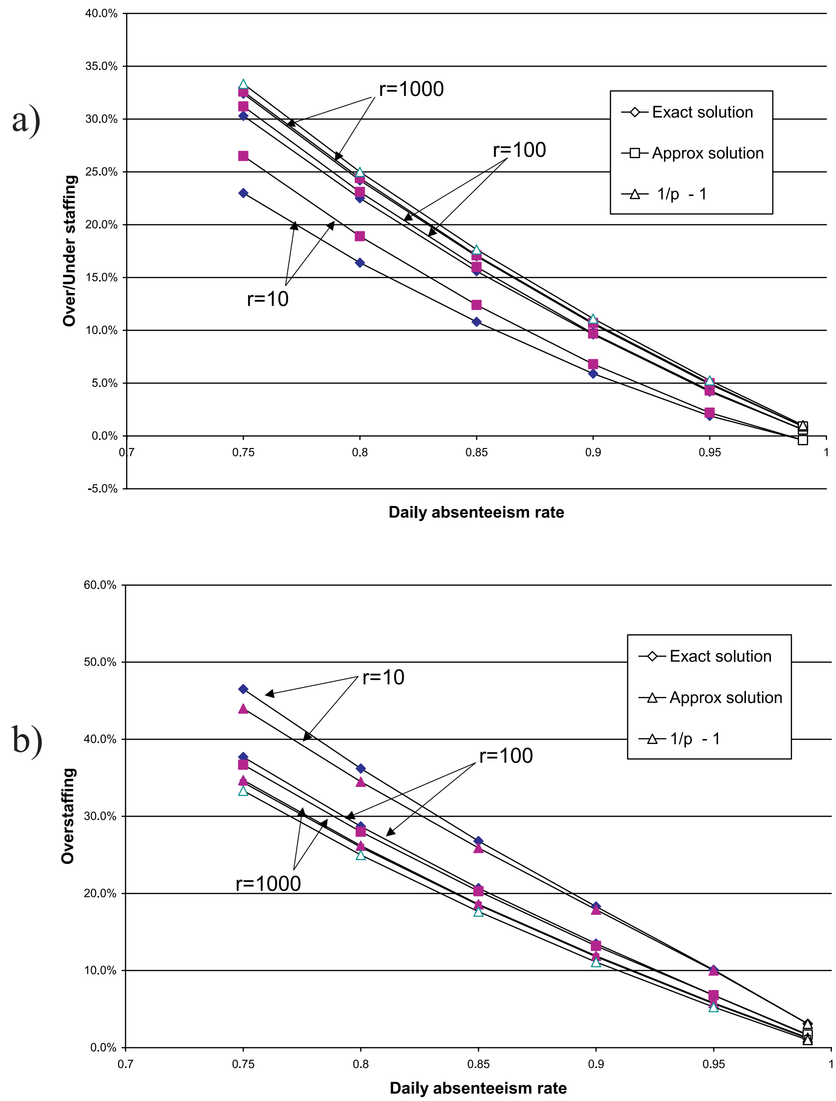

Figure 2.18: Optimal overstaff for different number of active tasks and show-up rates. The exact and approximated estimation are shown. a) low trippers wage: $w_t = 1.5w$, b) high trippers wage: $w_t = 4w$

2.5.2 Formulation for a full day: the time-dependent case

In this section $v(t)$ depends on time. Thus, a target curve $(n^*(t))$ is needed indicating the desired number of shifts to be scheduled at each moment t. This target is obtained from $v(t)$ using the time-independent formula (2.13). Then, drivers are hired to match this target as close as possible.

As an illustration, figure 2.19 shows two curves: the active task curve, $v(t)$ from figure 2.6 and its respective target curve, $n^*(t)$, assuming p=0.8 and $w_t = 3w$.

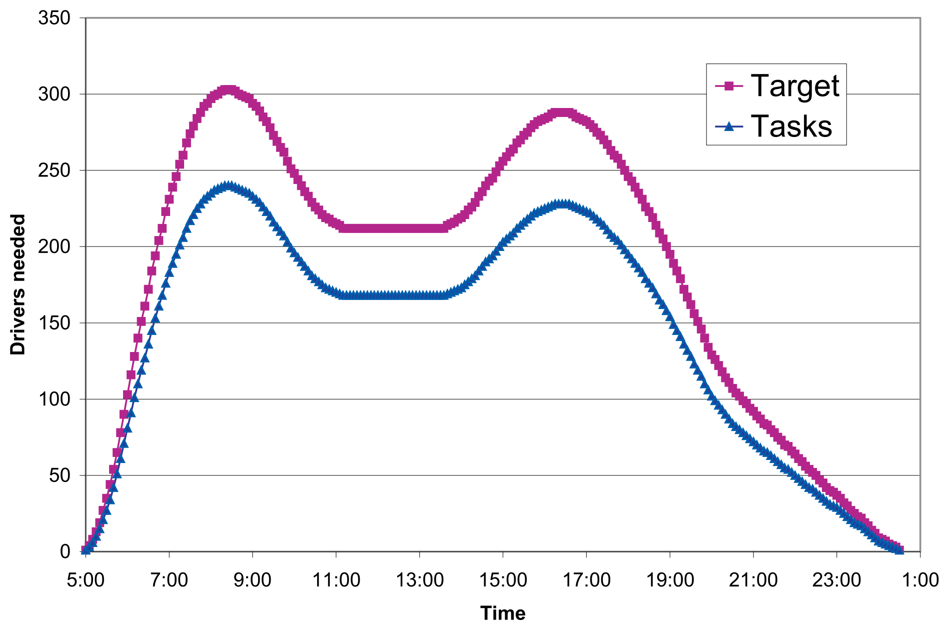

Figure 2.19: Daily target curve obtained from the active task curve based on equation (2.13)

Now the objective is to find a set of intervals that fits the target curve as close as possible. This is a deterministic shift filling problem just as in the deterministic case. The only difference is that in the deterministic problem the intervals need to cover the target curve, whereas now they only need to be close to it. To determine an optimal set of contracts, the expected cost of deviating from the target curve needs to be estimated. Such an expression should balance the extra costs of overshooting

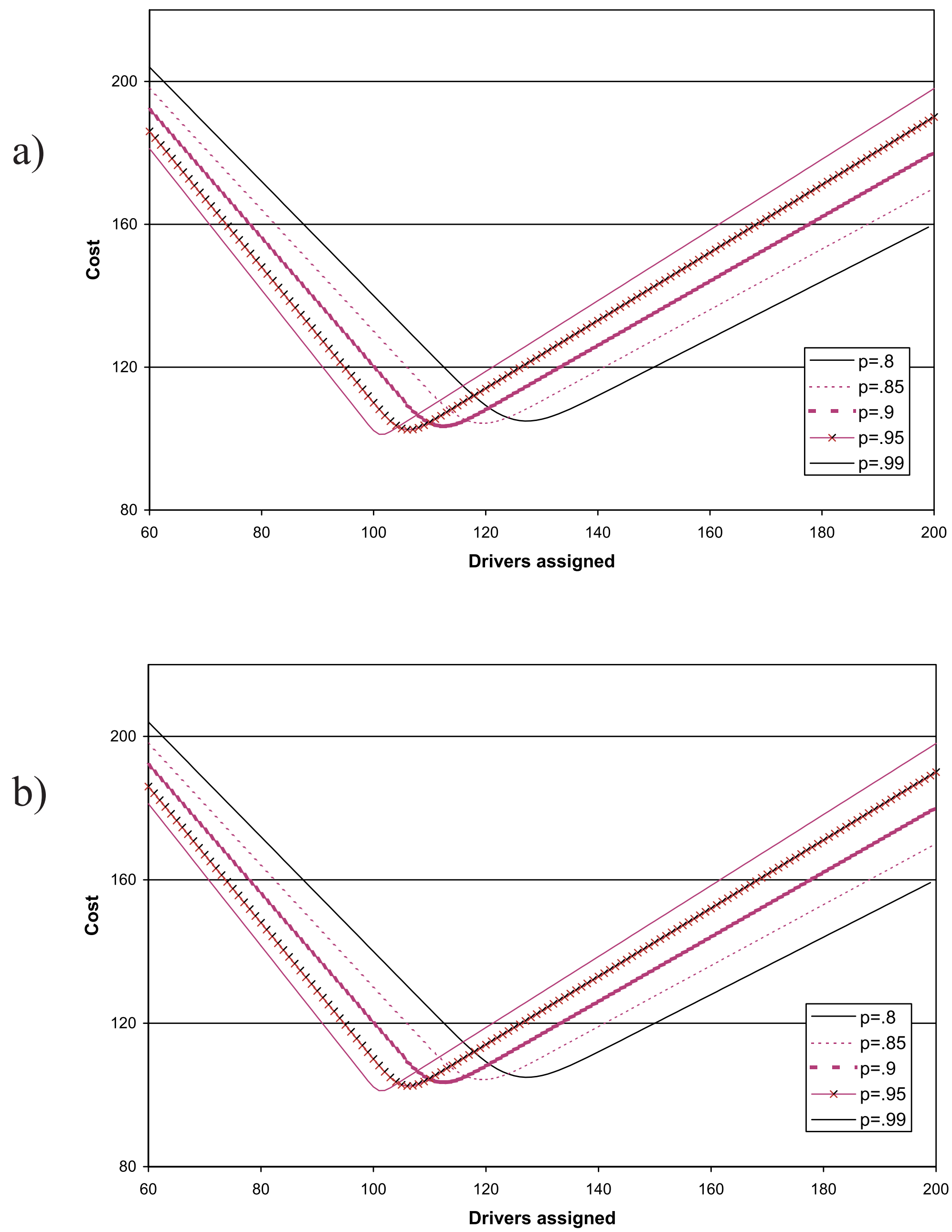

Figure 2.20: Optimal single time driver assignment cost for $v = 100$ and different absenteeism rates; a)$w_t = 1.5w$, b)$w_t = 3w$

and undershooting the target.

An exact formula for this cost is hard to obtain. Instead, approximations for this function are developed. Figure 2.20a and b show the cost of scheduling different numbers of drivers for the time independent case. In these cases, $v = 100$ active tasks are needed and different absenteeism rates are examined. These figures consider a tripper wage of 50% and 200% more than for regular drivers, respectively. From these curves and from others developed for $v = 40$ the following can be observed:

Observation 1 The cost of deviating from the target grows with the size of the deviation. Thus, minimizing the sum of the deviations (as in problem (2.14)) can be a good way of finding an assignment close to the target. However, if the solution to this problem is not unique then an assignment that minimizes the maximum deviation should be preferred.

Observation 2 The curves have a parabolic shape close to the minimum but they are asymptotically linear for medium deviations from the minimum. The slopes of these lines are easy to identify. Consider two extreme cases:

1. If no drivers are assigned then the cost is vw_t. If then a first driver is hired then the cost drops to $(v-1)w_t + w$ with probability p, and vw_t with probability $1-p$. This yields an expected cost of $vw_t - p(w_t - w)$ or a reduction on the total cost of $p(w_t - w)$.
2. If the system is heavily overstaffed, adding a driver increases the costs on w if that driver comes to work (probability p) and does not affect the

total cost otherwise. Therefore, the expected cost increases in wp.

Thus, deviations from the target should be weighted according with the derived rates: $(w_t - w)$ for understaffing deviations and w for overstaffing (p can be simplified from both sides when weighting the deviations).

Observation 3 Although the rate at which the penalties grow when the size of the deviations grow is independent of the value of v, the shape of the convex curve around the target is not. Indeed, the smaller v the bigger the penalty for a given deviation from the target. This suggests that instead of minimizing the sum of the deviations, the sum of the deviations divided by v could yield an assignment that provides a lower expected cost.

Observation 4 As p tends to 1 the curve takes a sharper "V" shape. In the case when $p = 1$, minimizing the sum of the deviations weighted according to Observation 2 is equivalent to solving the deterministic problem. Thus, in the limit when $p \to 1$ the approaches for the deterministic and stochastic problems converge.

If the combination menu is flexible enough to allow solutions that deviate only slightly from the target, then minimizing the sum of the deviations, the maximum deviation or other objective function aiming at matching the target curve as close as possible will provide solutions with very similar expected costs. However, if the combination menu is not flexible (or rigid) and deviations are always significant, then minimizing the sum of the deviations is used as a starting point, but different objective functions based on observations 1, 2 and 3 must be tested. These cases are analyzed in detail in the section 2.6.2.

Thus, the following modification of problem (2.5) is suggested to find an assignment that matches the target (in this case minimizing the area between both curves) as closely as possible:

$$min \sum_{t=1}^{T} \delta_t \tag{2.14a}$$

subject to

$$\delta_t \geq n^*(t) - \sum_{s \in S, f \in F(s)} \sum_{u=t-l_f+1}^{t} \theta_{fs}^u \qquad \forall t \in [1,T] \tag{2.14b}$$

$$\delta_t \geq -n^*(t) + \sum_{s \in S, f \in F(s)} \sum_{u=t-l_s+1}^{t} \theta_{fs}^u \qquad \forall t \in [1,T] \tag{2.14c}$$

$$\theta_{fs} = \sum_{t=1}^{T} \theta_{fs}^t = \frac{\gamma_f^s}{b_s} \sum_{t=1}^{T} \sum_{f \in F(s)} \theta_{fs}^t = \frac{\gamma_f^s}{b_s} \theta_s \qquad \forall f \in F(s), \forall s \in S \tag{2.14d}$$

$$\sum_{f \in F(s)} \theta_{fs} = \theta_s = b_s z_s \qquad \forall s \in S \tag{2.14e}$$

$$\theta_{fs}^t \in \aleph \qquad \forall f \in F(s), \forall s \in S, \forall t \in [1,T] \tag{2.14f}$$

$$z_s \in \aleph \qquad \forall s \in S \tag{2.14g}$$

$$\delta_t \geq 0 \qquad \forall t \in [1,T]. \tag{2.14h}$$

The only difference between this formulation and problem (2.5) is in the objective function and in the task covering constraints (constraints (2.14b) and (2.14c) replace constraints 2.5b). In this formulation δ_t represents the absolute value of the deviation from the target at time t. The objective function consists in minimizing the sum of these deviations [11]. Constraints (2.14b) and (2.14c) connect the value of δ_t with the size of the deviation. Notice that the combination feasibility constraints remain the same.

Unlike the deterministic case, the solution of problem (2.14) is feasible for the shift filling problem (any interval assignment satisfying the combination feasibility constraints would be). However, it might not be optimal. Thus, an upper bound for the minimum expected cost can be obtained by simulating the performance of the solution over the planning horizon. A lower bound can be obtained by evaluating (2.7) for $n = n^*(t)$ and then add across all intervals[12] as follows:

$$\int_{t=0}^{T} \left[wpn^*(t) + w_t \int_{x=-\infty}^{v(t)} \phi(x; n^*(t)p, n^*(t)p(1-p))(v(t)-x)dx \right] dt \qquad (2.15)$$

The tests developed later in this dissertation show that the gap between these bounds is narrow for planning purposes.

In this section, the target curve was identified assuming that all combinations were equally paid per hour. In the next section, the case where combinations are

[11] An alternative objective function, in which the cost of deviating from the target is estimated using a second degree polynomial expansion, is developed in Appendix B. However, as will be shown, the simpler objective (2.14a) provides very narrow gaps.

[12] Since this would be the optimal solution for the problem if the length of the shifts worked by all drivers were equal to the length of the discretization interval.

paid at different rates is considered.

2.5.3 Different wages

The target curve procedure developed in section 2.5.1 assumed that all combinations were paid equally per hour worked. However, in the heterogenous wage case the target curve can not be identified a priori since the procedure balances the wage of regular drivers and trippers. And in this case the wage of regular drivers at each moment depends on the output of the shift filling problem. If the average wage of shifts scheduled at t changes, so should the target curve $n^*(t)$. Thus, the target curve depends on the optimal solution for the problem. This makes the problem non-linear and harder to formulate. Thus, the formulas developed in the previous section can not be used directly here.

To solve this problem the following iterative approach is suggested:

Step 1. Obtain $n * (t)$ as if all combinations were paid equally with a wage w (e.g. choose the average (or the lowest) wage among combinations in the menu).

Step 2. Given $v_t = n^*(t)$ Obtain the optimal solution for problem (2.14).

Step 3. For the obtained solution compute $w(t), \forall t$, where $w(t)$: the average wage paid to all drivers working at time t.

Step 4. Obtain $n^*(t)$ for each time slice as in section 2.5.1 using $w(t)$ as the wage paid to regular drivers for each time slice.

Step 5. If two consecutive iterations yield similar $n^*(t)$ or yield a similar total cost, stop. Otherwise go to step 2.

Once the target curve, $n^*(t)$ is obtained, the best-fitting combinations should be obtained similarly to the homogeneous wage case. Notice that the heuristic algorithm does not guarantee convergence to the optimal target curve.

In order to test the rate of convergence, the procedure was applied over the same active tasks as in previous sections (see figure 2.6) with the following combination menu:

Combination 1 of table 2.1: Eight hours a day, five-days a week

Combination 4 of table 2.2: Three days working ten hours and two days only five.

Combination 8: One split shift of four hours in the morning and four hours in the evening five days a week.

The first two combinations were paid at \$12 per hour while the last one was paid at \$18 per hour. In this case, the procedure appeared to be very effective converging to a target curve after two iterations.

2.6 The Stochastic Problem: Validation

In section 2.4.1 the solution based on (2.5) was shown to be achievable (or very close) if enough flexibility is allowed in the combination menu for the deterministic case ($p = 1$). In this section the lower bound on the expected cost for the shift filling problem when drivers come to work with a probability p (based on (2.15)) is tested.

To test it, the lower bound is compared with the expected cost of the optimal solution of problem (2.14). This solution is feasible for the stochastic shift filling problem. The expected cost is estimated using a 100-day simulation. Within each day of the simulation, trips are assigned to intervals using the heuristic of Appendix A. In all these tests an absenteeism rate $p = 0.8$ (the approximation works better for higher absenteeism rates so this is a worst case scenario) and a tripper salary that tripled the regular salary are assumed. The active task curve (and its underlying trips) considered was, as before, the one on figure 2.6.

Two cases can be distinguished: when the assignment closely matches the target curve ($n^*(t)$) and when it does not. Of course, the lower bound should predict much better the total cost in the first case.

2.6.1 Solution slightly deviating from the target

To test the accuracy of the lower bound a simple combination menu that yielded a very close match to the target for the deterministic case was used:

Combination 1 of table 2.1: Eight hours a day, five-days a week

Combination 4 of table 2.2: Three days working ten hours and two days only five.

Combination 8: One split shift of four hours in the morning and four hours in the evening five days a week.

In this case, all three combinations are assumed to be paid equally. The procedure consists in finding a target curve for the timetable (using equation (2.13)) and its associated lower bound on the total expected cost (by replacing the target on function (2.15)); in this case, the procedure predicts a lower bound of $35,748.0. Then the shift filling problem is solved (using model (2.14)) yielding a list of intervals that are assigned to drivers according with the combinations. Then, the driver assignment (obtained from problem (2.14)) is simulated over 100 days (considering absenteeism). The simulation consists of determining the drivers actually working and assigning them to the trips of the schedule using the heuristic from Appendix A. This heuristic yields the total number of trips being worked by trippers and overtime. Thus, an operational cost for each day is obtained. For the 100 days, the average cost is $36,053.3 (with a standard deviation of $534.7) that is 0.85% over the prediction.

To test the quality of this methodology, two simple heuristics that mimic current practice are proposed as a basis for comparison. In the first one, a target $n(t)$ is determined by multiplying the active task curve by a fixed value p'. Different values of p' give different assignments. Each can be tested following the same procedure as before. The second heuristic mimics ACT's policy[13]: solve the deterministic problem

[13] Although ACT performance can be estimated better by looking at the results on section 2.6.2

and determine the minimum number of drivers to cover the active tasks. Then, add a fixed fraction of them ($p' - 1$) to deal with absenteeism. Their results are shown as “heuristic 1” and “heuristic 2”, respectively in figure 2.21.

The heuristics perform similarly. Notice that the optimum p' in both cases is approximately 1.28; slightly over $1/p = 1.25$. Since $w_t > 2w$ this result is consistent with the suggestion made at the end of section 2.5.1 (if $w_t > 2w$ then more than $1/p$ drivers should be scheduled per trip). Most important, notice that the solution derived from problem (2.14), denoted as a dashed line in the figure, performs slightly better than both heuristics even if the best p' is identified.

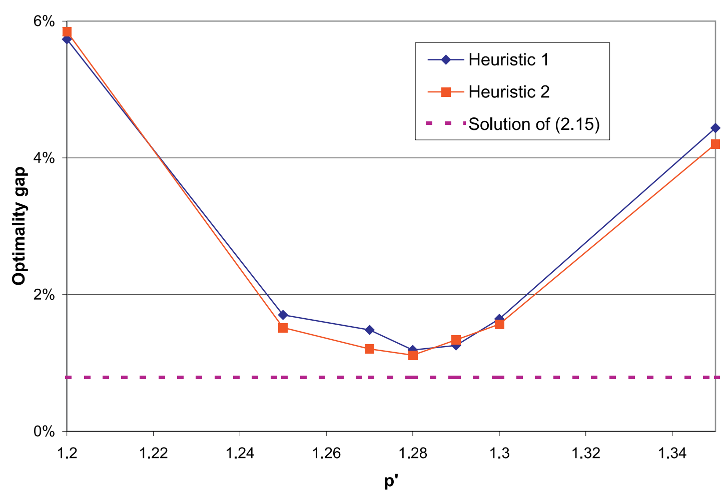

Figure 2.21: Gaps between the simulated expected cost and the lower bound for a flexible combination menu. Three strategies are displayed: heuristic 1 (for different p'), heuristic 2 (for different p'), and use the assignment of problem (2.14).

2.6.2 Solution grossly deviating from the target

In this section the case when the assignment does not match the target curve closely is analyzed. Alternative objective functions provide better solutions for the shift filling problem than minimizing the sum of the deviations from the target (as in problem (2.14)). In this section a set of different objective functions based on observations 1,2 and 3 of section 2.5.2 are suggested and tested.

Consider a more rigid combination menu:

Combination 1 of table 2.1: Eight hours a day, five-days a week.

Combination 2 of table 2.1: Four days working nine hours and one day working only four.

Combination 4 of table 2.2: Three days working ten hours and two days working only five.

Following the same procedure explained for the flexible menu case, the simulation for the solution obtained from problem (2.14) yields a total cost of $\$42,094.6$, 17.8% more than the $\$35,747.5$ of the lower bound (same as in the flexible menu case).

Then, the first observation of the list of section 2.5.2 is tested by modifying problem (2.14). Now, the objective value of problem (2.14) is added as a constraint and instead the largest δ_t is minimized. The simulation for the solution obtained from this second model yields a total cost of $\$40,299.1$.

Finally, the remaining observations (2 and 3) are tested together using the following formulation:

$$min \sum_{t=1}^{T} \delta_t / n^*(t) \tag{2.16a}$$

subject to

$$\delta_t \geq w \left[n^*(t) - \sum_{s \in S, f \in F(s)} \sum_{u=t-l_s+1}^{t} \theta_{fs}^u \right] \qquad \forall t \in [1, T] \tag{2.16b}$$

$$\delta_t \geq (w_t - w) \left[-n^*(t) + \sum_{s \in S, f \in F(s)} \sum_{u=t-l_s+1}^{t} \theta_{fs}^u \right] \qquad \forall t \in [1, T] \tag{2.16c}$$

$$\theta_{fs} = \sum_{t=1}^{T} \theta_{fs}^t = \frac{\gamma_f^s}{b_s} \sum_{t=1}^{T} \sum_{f \in F(s)} \theta_{fs}^t = \frac{\gamma_f^s}{b_s} \theta_s \qquad \forall f \in F(s), \forall s \in S \tag{2.16d}$$

$$\sum_{f \in F(s)} \theta_{fs} = \theta_s = b_s z_s \qquad \forall s \in S \tag{2.16e}$$

$$\theta_{fs}^t \in \aleph \qquad \forall f \in F(s), \forall s \in S, \forall t \in [1, T] \tag{2.16f}$$

$$z_s \in \aleph \qquad \forall s \in S \tag{2.16g}$$

$$\delta_t \geq 0 \qquad \forall t \in [1, T]. \tag{2.16h}$$

Again, the only differences between this model and the previous one (2.14) are in the objective function and in constraints (2.16b) and (2.16c). Notice how each element of the sum of the variables in the objective function is weighted by the inverse of the target for each time t. Additionally, the size of the penalties are weighed differently for understaffing and overstaffing conditions (constraints (2.16b) and (2.16c)). The simulation for the solution obtained from this second model yields a total cost of $\$40,179.5$ reducing further the upper bound for the problem.

It should be noted that identifying the best combination from the list of observations in section 2.5.2 is a heuristic procedure and might depend on each particular problem. However, the tests in this research suggest problem (2.14) is suitable for flexible combination menus while problem (2.16) is better for rigid ones.

To test the quality of the solution obtained, the same two heuristics proposed in the previous section are tested. Their results are shown as "heuristic 1" and "heuristic 2" in figure 2.22.

In this case the heuristics provide a similar cost but using different values for p'. Again, notice that the heuristics are outperformed by the procedure derived in section 2.5.2 and that more than 10% of the total driver cost is added due to the rigidity of the combination menu.

2.7 Policy Implications

This chapter illustrates the potential of the design tools proposed in this research by examining how absenteeism rates and tripper wages affect costs.

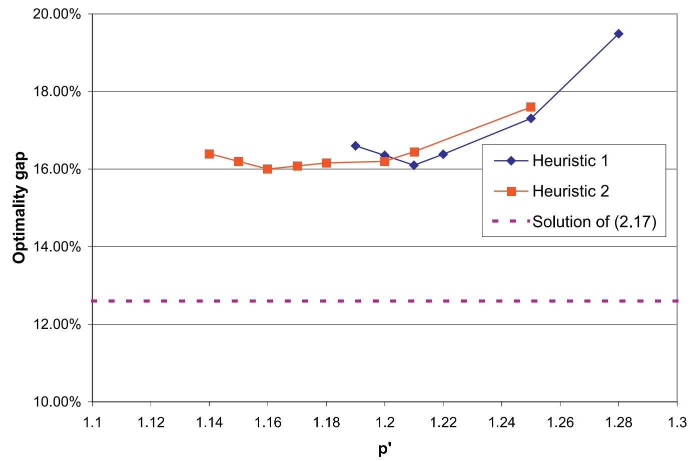

Figure 2.22: Gaps between the simulated expected cost and the lower bound for a rigid combination menu. Three strategies are displayed: heuristic 1 (for different p'), heuristic 2 (for different p'), and use the assignment of problem (2.16).

Figure 2.23a shows the impact of different absenteeism rates on the total driver cost for the two combination menus analyzed in the previous section. They are called flexible and rigid, respectively. The figure includes both the upper bound obtained from the 100-day simulation and the lower bound from the continuous approximation. In the flexible case, the gap between the upper and lower bound remains approximately constant around 1% suggesting that impacts in the optimal solution can be (quite accurately) estimated using the approximated solution. Since the approximated solution is governed by a closed formula, planners could use this formula to predict the impact of their policies. In the case when the combination menu is rigid, the gaps are considerably larger but still approximately constant. This correlation between bounds indicates that agencies could still estimate the impact of absenteeism rate on the optimal solution by looking at the impact on the lower bound.

Agencies prefer smooth operations where days do not differ much from each other. Therefore, an operation where the number of trippers needed changes dramatically over days is not desirable. In this regard, figure 2.23b shows the expected cost in trippers and overtime and its standard deviation for different absenteeism rates in both combination menu cases. In both cases the standard deviation and the expected cost grow approximately linearly with the absenteeism rate. When the combination menu is rigid, considerably more trippers are used. The standard deviation in both cases are not significantly different.

Analogously, the impact of trippers wage on the total cost is illustrated in figures 2.24a and 2.24b. As expected, the cost grows with the trippers wage and in the case of flexible combination menus the rate of changes of the upper and lower bounds on the total cost are highly correlated. However, when the combination menu is rigid, the gap between lower and upper bound tends to grow with the trippers wage. Although trippers cost grow with trippers wage, when converted into number of trippers (dividing each point in these plots by the trippers wage) the tendency is reverted, that is fewer trippers are required.

An additional policy tool that can be obtained from this analysis is the marginal driver cost of a trip. Very often agencies are faced with the tradeoff between increasing their peak frequencies to increase their riderships or reduce them to save some resources that are mostly inactive during the rest of the day. Although this dissertation does not analyze this issue in depth, the tools developed in this chapter can be used to identify the effect on drivers cost of trimming or increasing frequencies.

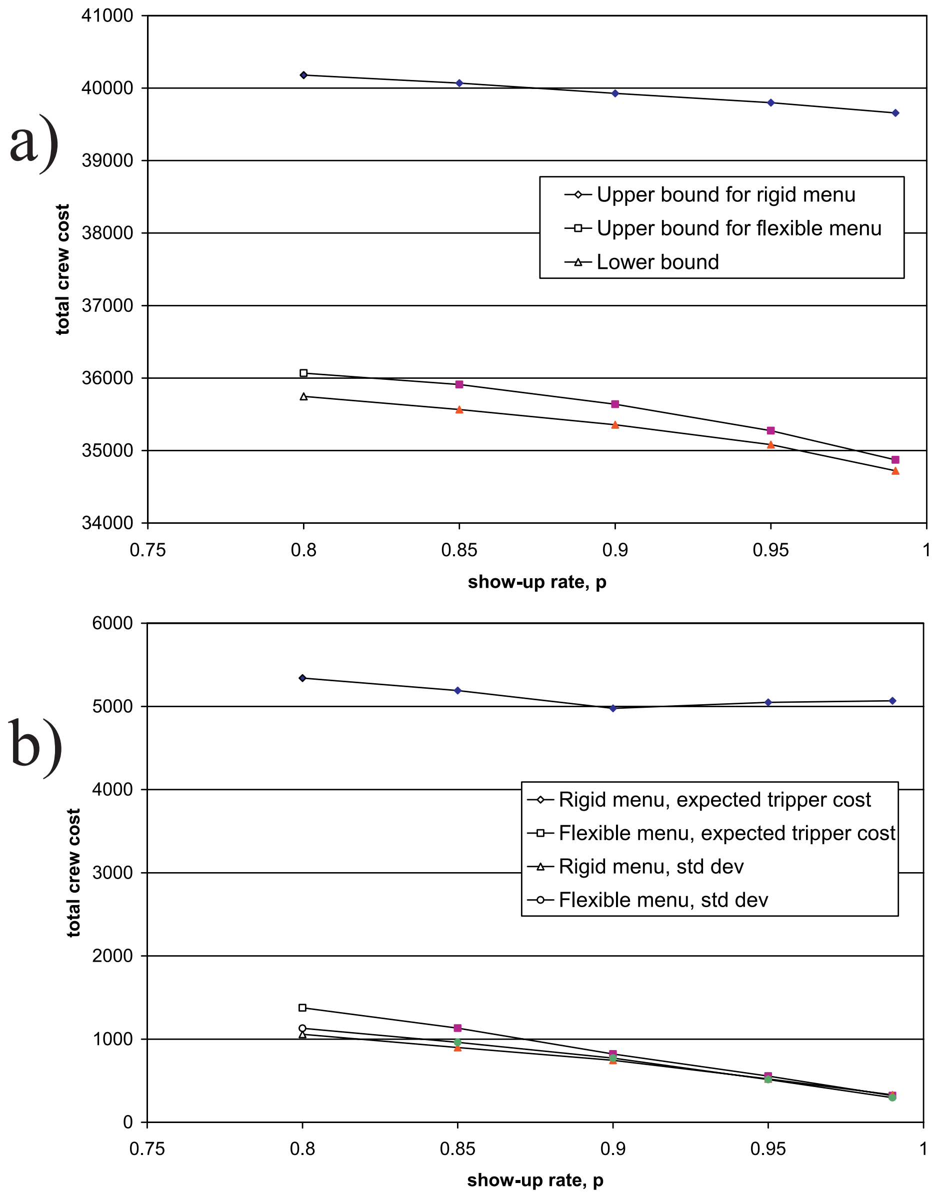

Figure 2.23: Impact of absenteeism in optimal driver assignment; a) in total expected driver cost, b) in tripper cost and standard deviation.

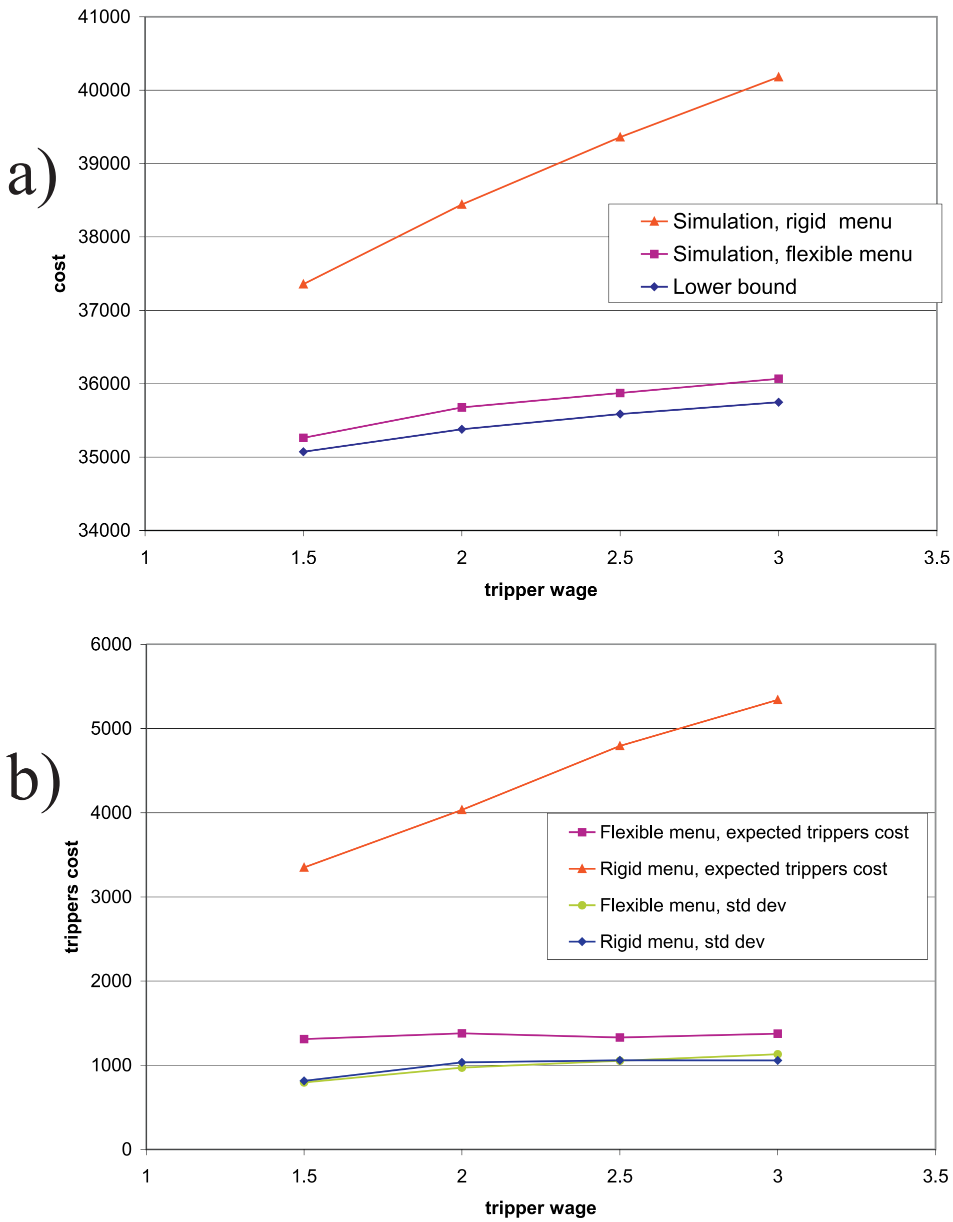

Figure 2.24: Impact of tripper wage in optimal driver assignment; a) in total expected driver cost, b) in tripper cost and standard deviation.

This chapter has provided tools to solve the shift design problem considering absenteeism. The shift-driver design has been studied and different designs have been evaluated. The shift filling problem has been formulated, bounds have been provided and tests have been shown. These tools should be valuable for transit agencies since they address this problem frequently (e.g. ACT redefines its shifts every three months). Although less frequently, agencies also deal with a harder problem. That problem consists in defining not only the shift sequences to be offered but the timetables to be run by each line. The next chapter is devoted to this goal.

Chapter 3

Joint Shift-Timetable Design Problem

3.1 The Problem

The joint shift-timetable design problem consists of determining the timetable for each line and the contract (shift sequence and wage) for each driver simultaneously to minimize the expected cost over a design period.

Since most transit agencies are publicly owned, all costs or impacts should be considered in system design. These impacts include the cost of operating the vehicles, buying or leasing the fleet, paying the drivers, and waiting by the passengers. Regarding vehicle operation a fixed cost per mile driven is assumed. Since the lines' length is constant this implies that the cost per roundtrip-line is also considered fixed. Labor is valued by salaries depending on the work length and characteristics (e.g. split shifts, overtime, tripper, etc.). The impact of passenger waiting time is assumed proportional to the total passenger delay.

The two decision variables of this problem are dispatching schedules for vehicles and shift driver assignments. The fleet size is assumed to be sufficient to accommodate all trips, and its characteristics (size, capacity, speed, operation cost, etc.) are assumed given.

Two uncertainties are considered: absenteeism and passenger demand. Although both uncertainties affect the design process, the agency can react on the daily operation only to absenteeism. Timetables can not be altered on a daily basis and therefore the operator can not react to passenger demand. The operator can control the driver-trip assignment, pay overtime and call trippers to address absenteeism.

In the next section the joint shift-timetable design problem for the single-line case is introduced. This simple case helps to understand and graphically visualize the relations between different components of the system.

3.2 Illustration

Consider the deterministic case of a single line transit system over one day of operation. For clarity, in this section variables are defined using lower case letter and data using capital letters. Call

$A_p(t)$: cumulative number of passengers willing to take a bus by time t.

$a_c(t)$: cumulative number of intervals that have been started by time t.

$d_c(t)$: cumulative number of intervals that have been ended by time t.

$d_p(t)$: cumulative number of passengers that have boarded a bus by time t.

$a_b(t)$: cumulative number of vehicle trips dispatched by time t.

$d_b(t)$: cumulative number of vehicle trips finished by time t.

Figure 3.1 graphically illustrates a feasible solution using these six curves. Finally:

K: capacity of the vehicles.

R: roundtrip duration of the line.

$v(t)$: number of active tasks at time t.

$n(t)$: number of active intervals at time t.

The design goal is to minimize the sum of the total delays suffered by the passengers, the resources involved in operating the vehicles and the salaries paid to drivers weighed by some parameters. In figure 3.1 the delays can be estimated as the area between $A_p(t)$ and $d_p(t)$. The operational cost is proportional to the total number of vehicles dispatched during the whole day ($a_b(T)$). The hours spent by drivers are equal to the sum of all unitary horizontal time slices between $a_c(t)$ and $d_c(t)$ [1]. The weights given to each of these impacts are a political decision that the transit

[1] Since intervals are not finished in the same order that they are started, each horizontal slice does not correspond to the work done by a single driver. Therefore, more information is needed to determine the total salaries paid.

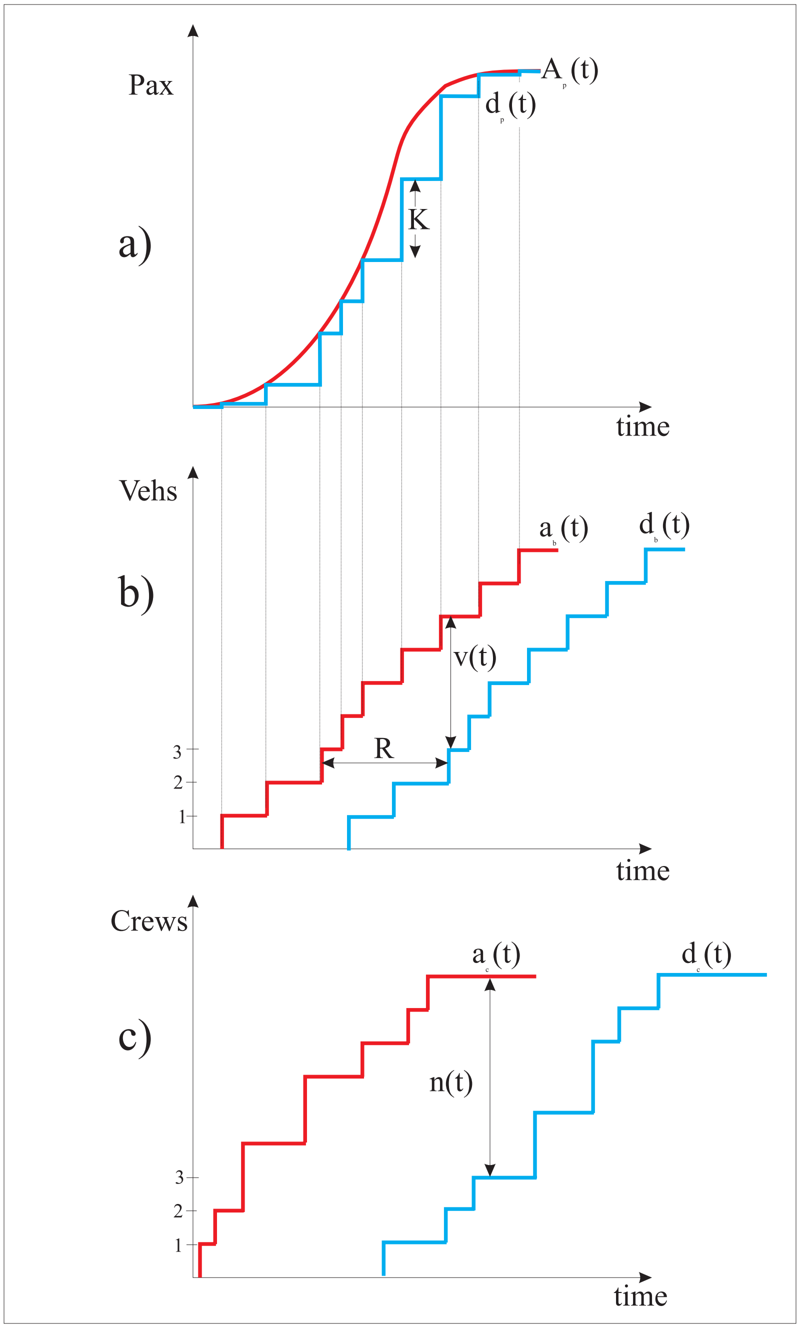

Figure 3.1: Graphical illustration of a feasible solution to the problem; a) cumulative arrival and departure of passengers to the hub, b) cumulative departure and arrival of vehicles to the hub, c) cumulative number of intervals to have started and ended

agency must make.

Additionally, some relations and constraints must be satisfied. Once the curve $a_b(t)$ is determined, the curves $d_b(t)$, $d_p(t)$ and $v(t)$ are determined as well. The curve $d_b(t)$ is defined since every time a vehicle departs, it will return after R units of time,

$$d_b(t) = a_b(t - R)$$

and every time a vehicle departs, it takes as many passengers as possible from the hub. Thus, $d_p(t)$ is immediately defined. For a small ϵ,

$$d_p(t) = \begin{cases} d_p(t-\epsilon) + min(K, A_p(t) - d_p(t-\epsilon)) & \text{if a vehicle is dispatched at } t, \\ d_p(t-\epsilon) & \text{otherwise.} \end{cases}$$

Regarding vehicle availability, the size of the fleet, N must be bigger than $v(t)$ $\forall t$. Regarding drivers, the active tasks can not exceed the active intervals, that is :

$$a_b(t) - d_b(t) = v(t) \leq n(t) = a_c(t) - d_c(t)$$

Finally, vehicles can change drivers only at the hub.

Therefore, although the three dimensions of this problem (passengers, vehicles and drivers) are linked, the system and its cost is totally defined once the timetable $(a_b(t))$ and the contracts $(a_c(t)$ and $d_c(t))$ are determined.

Because $a_b(t)$ is a step function, an optimal solution for this problem can be achieved using integer programming. This approach is useful to identify the best (or at least a good) solution for the deterministic problem. However, the combinatorial nature of this approach prevents a closed form cost function indicating the impact of each parameter in the total cost. Furthermore, these tools are of little value when uncertainties are considered. Thus, combinatorial optimization might provide good tools to operate the system (control mechanism), but alternative approaches are needed for design.

Chapter 2 shows that with a flexible combination menu a target curve can be matched very closely without using split shifts. This simplifies the deterministic joint shift-timetable design problem considerably since all drivers work the same number of weekly hours for the same salary. The driver cost associated with each trip can be assumed proportional to the length of the trip. In this case, this joint design problem becomes considerably more simple since the operational cost of each trip can include the driver cost. Then the joint shift-timetable design problem can be solved in two steps: first identify the optimal timetables assuming unconstrained drivers but adding the driver wage as part of the operational cost, and second solve the shift design problem to cover those timetables.

The stochastic joint shift-timetable design problem also becomes simpler if a flexible combination menu is considered. However, the driver cost associated with a trip is not as straightforward since the number of regular drivers scheduled per trip is now more than one and eventually trippers or overtime may be needed. Thus, the driver cost per time of a trip is strictly greater than the wage of a regular driver if

$p < 1$. This extra cost is plotted in figures 3.2a and 3.2b versus different number of active tasks and absenteeism rates for $w_t = 1.5w$ and $w_t = 4w$, respectively. The two steps procedure suggested for the deterministic case can be tried here assuming a certain (time dependant) cost per trip. If the expected cost per trip of the solution differs from the original assumptions a new iteration of the two-steps procedure may be needed using the updated expected costs per trip.

If the combination menu is not flexible enough or if all combinations are not equally paid, then the (expected) cost per trip can not be predicted since it depends on the output of the shift design problem (the time dependant idle hours and combination of different shifts). In these cases, an iterative approach between the shift filling problem and the *timetable design problem* is followed. In the timetable design problem it is assumed that the shift sequences are already defined, and the timetables per line need to be determined. This problem is tackled in following sections.

The timetable design problem is solved first using continuum approximation and assuming no uncertainties. Later in this chapter uncertainties are included. To test the accuracy of the approximations, the deterministic problem is solved using mixed integer mathematical programming in chapter 4.

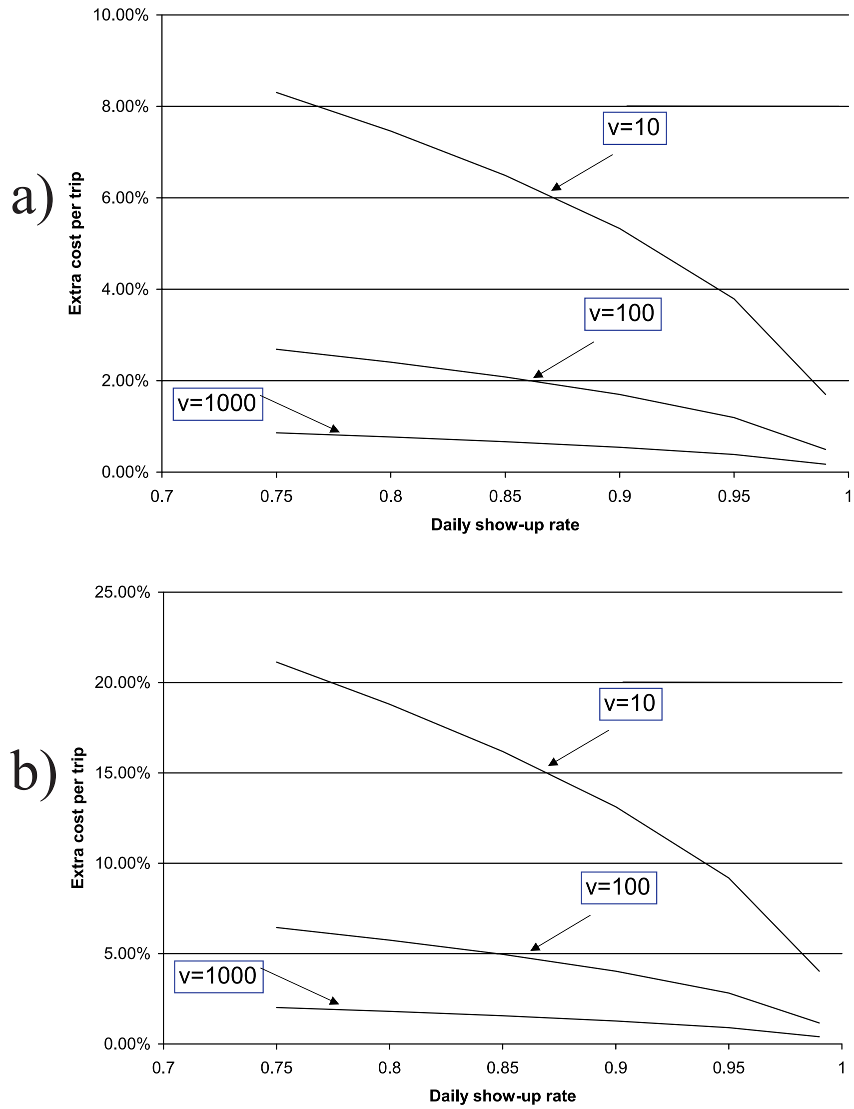

Figure 3.2: Extra cost per trip for different number of active tasks and absenteeism rates a) low trippers wage: $w_t = 1.5w$, b) high trippers wage: $w_t = 4w$.

3.3 Timetable Design: the Deterministic Problem

In the timetable design problem, routes, fleet, driver-shift pairs and passenger demand are assumed to be known. However, the timetable followed on each route and the assignment between drivers and trips must be determined. In reality, solving this problem might be needed to fine-tune the system operation; e.g. to adjust the service to better meet the updated demand prediction. In this case, the agency might not want to alter the routes or affect the drivers contracts. The most inexpensive alternative might be to update the timetable. Or equivalently, imagine that the agency will face a short period of time when the demand will be very different than for the rest of the year and when timetables should be modified. This problem is also important as a building block for the joint shift-timetable design problem (timetables and shifts being designed simultaneously).

The problem is solved in two stages: initially it is assumed that drivers are never absent and that passenger demand is deterministic. In a second stage, absenteeism and stochastic demand are considered. In the deterministic problem the objective is to minimize the sum of operational and waiting costs constrained to driver availability. In the stochastic problem a control strategy is needed to assign trips to drivers and then deal with understaffed periods of the day. In this case the expected cost must also consider drivers' salaries.

Since the focus of this dissertation is on design issues some simplifications are made. Consider a passenger arriving at time t at stop i in line 1, with stop j of line 2 as her/his destination. This passenger has to wait twice: at bus stop i and at the hub during the transfer. Let R_1^i be the travel time from the hub to stop i. If instead the passenger is modelled as having her/his origin at the hub at time $t - R_1^i$, then her/his waiting time at the bus stop will be (correctly) computed at the hub. This simplification allows one to compute all the origin-based waiting times at the hub and convert the problem into a one-to-many distribution problem. Two problems remain:

1. The time that this passenger arrives to the hub to transfer lines is a function of the output of the model. However, in this model the timetables are not synchronized so it can be assumed that the passenger arrives at the hub at time $t + R_1 - R_1^i$, where R_1 is the roundtrip duration of line 1.

2. The maximum load for any bus departing from the hub also depends on the timetable. It is assumed that the maximum load is always equal to the total number of passengers entering the bus at the hub. Notice that since the trip origin is shifted to the hub, the maximum load assumed for a trip corresponds to the total boardings of the trip.

These simplifications allow one to treat the problem as if its demand was one-to-many (all trips generating at the hub). The next section presents a mathematical model for the deterministic timetable design problem.

3.3.1 Formulation

Since dispatching vehicles is inherently discrete, formulating this problem using mathematical programming requires binary variables, and solving requires integer programming tools. The moments when vehicles can be dispatched are defined by discretizing time in small intervals of length δ (δ being small, for example five minutes). Without loss of generality, δ is assumed to be 1 (so time is measured in units of δ).

A feasible assignment can not assign a trip to an interval if the trip starts before the interval or ends after the interval. Also, overlapping trips can not be assigned to the same interval.

Additionally, passengers can ride a bus only when a vehicle is dispatched. The batch of passengers riding a bus can not exceed the capacity of the bus or the number of passengers waiting at the bus stop.

Let start defining the relevant sets and data:

J : the set of intervals being worked.

b_j : the time interval j begins.

e_j : the time interval j ends.

L : the set of lines in the system.

R_l : the roundtrip duration over line l in δ units of time.

α : penalty assumed for a passenger waiting δ units of time.

β : cost paid for a bus operating for δ units of time.

A^l_{pt} : cumulative number of passengers that would like to take line l by time t.

K : capacity of a vehicle.

and the following variables:

$$x_t^{jl} = \begin{cases} 1 & \text{if a trip on line } l \text{ departing at time } t \text{ is assigned to interval } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$x_t^l = \begin{cases} 1 & \text{if a trip on line } l \text{ departs at time } t \\ 0 & \text{otherwise.} \end{cases}$$

$$y_t^l = \text{number of passengers starting their trip on line } l \text{ at time } t$$

Notice that $x_t^l = \sum_{j \in J} x_t^{jl}$ and that y_t^l can only be positive if $x_t^l = 1$. Now, the problem can be formulated using mathematical programming as follows:

$$\min \sum_{l \in L} \sum_{t=1}^{T} \left\{ \alpha \left(A_{pt}^{l} - \sum_{v=1}^{t} y_{v}^{l} \right) + \beta R_{l} x_{t}^{l} \right\} \tag{3.1a}$$

subject to

$$x_{t}^{jl} + x_{t+\Delta}^{jm} \leq 1 \qquad \forall j \in J, \forall \Delta < R_{l}, \forall m \in L; \tag{3.1b}$$

$$\text{if } m = l \text{ then } \Delta > 0$$

$$x_{t}^{jl} = 0 \qquad \forall j \in J, \forall l \in L, \forall t < b_{j} \text{ or } t > e_{j} - R_{l} \tag{3.1c}$$

$$y_{t}^{l} \leq K x_{t}^{l} = K \sum_{j \in J} x_{t}^{jl} \qquad \forall j \in J, \forall l \in L, \forall t \in [1, T] \tag{3.1d}$$

$$\sum_{v=1}^{t} y_{v}^{l} \leq A_{pt}^{l} \qquad \forall l \in L, \forall t \in [1, T] \tag{3.1e}$$

$$x_{t}^{jl} \in \{0, 1\} \qquad \forall t \in [0, T], \forall j \in J, \forall l \in L \tag{3.1f}$$

$$y_{t}^{l} \geq 0 \qquad \forall l \in L, \forall t \in [1, T]. \tag{3.1g}$$

The objective function consists of two elements. The first is the sum of passenger waiting time weighed by a parameter α indicating the agency's value of the passenger waiting time. The second element corresponds to the operating cost. Determining α is a political decision and the results may be highly sensitive to its value. However, it is widely used among transit agencies since it solves the difficult trade-off between impacts measured with different units. Constraints (3.1b) prevent overlapping trips from being assigned to the same interval. Constraints (3.1c) forbid the assignment of trips to an interval if they can not be completed during the interval. Constraints (3.1d) restrict the number of passengers boarding a vehicle to be under the capacity of the vehicle. Constraints (3.1e) guarantee the cumulative number of passengers to have boarded vehicles by time t in each line does not exceed the cumulative demand for the line by that time.

Notice that due to the discretization of time in this model the waiting times are miscalculated. Passenger waiting times during the period in which they arrive at the stop and during the period in which they board a vehicle are ignored. Therefore, the objective function underestimates the total waiting cost by $\sum_{l=1}^{L} \frac{u_T^l}{2}$ (which is a constant for this problem).

This problem is quite complicated and solving it requires integer programming tools. Unfortunately, such a process would neither provide a closed formula to identify the effect of each of the parameters in the total cost nor allow the exploration of the absenteeism effects in the optimal design. Thus, the same approximation used in the previous chapter is used here to relax problem (3.1). Assuming that intervals (i.e., drivers) can shift trips infinitely fast, a simpler model emerges. This simplified model provides a lower bound for problem (3.1), and is more friendly to consider uncertainties. Furthermore, a closed form solution for the problem based on continuum approximations is obtained.

3.3.2 Simplified formulation

Let $n(t)$ be the number of active intervals at time t and consider the same variable set as in problem (3.1):

$$x_t^l = \begin{cases} 1 & \text{if a vehicles starts a trip on line } l \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the model can be reformulated as follows:

$$\min \sum_{l \in L} \sum_{t=1}^{T} \left\{ \alpha \left(A_{pt}^l - \sum_{v=1}^{t} y_v^l \right) + \beta R_l x_t^l \right\} \tag{3.2a}$$

subject to

$$\sum_{l \in L} \sum_{i=t-R_l}^{t} x_i^l \leq n(t) \qquad \forall t \in [1 \ldots T] \tag{3.2b}$$

$$y_t^l \leq K x_t^l \qquad \forall l \in L, \forall t \in [1, T] \tag{3.2c}$$

$$\sum_{v=1}^{t} y_v^l \leq A_{pt}^l \qquad \forall l \in L, \forall t \in [1, T] \tag{3.2d}$$

$$x_t^l \in 0, 1 \qquad \forall t \in [1, T], \forall l \in L \tag{3.2e}$$

$$y_t^l \geq 0 \qquad \forall l \in L, \forall t \in [1, T]. \tag{3.2f}$$

The objective function and the sets of equations (3.2c) and (3.2d) are the same as in problem (3.1). However, constraints (3.1b) and (3.1c) are replaced by constraints (3.2b). These constraints require that the number of active tasks must not exceed the number of active intervals at any instant of time.

However, as was discussed in chapter 2, constraints (3.2b) are necessary but not sufficient for the solution to be feasible. Thus an optimal solution for problem (3.2) provides a lower bound for problem (3.1).

This problem is first studied using a continuum approximation. This tool yields a cost function that shows how the different elements of the problem influence cost. Later, following this approach, uncertainties are considered. Finally, to test the approach, the problem is solved optimally with numerical optimization and cutting planes.

3.3.3 Continuum Approximation

In this section the problem is formulated using the nomenclature from section 3.2 but extended to many lines. Let L be the number of lines, K_l the capacity of vehicles serving line l, R_l the roundtrip duration of line l, and:

$A^l_p(t)$: cumulative number of passengers willing to take the line l by time t.

$A_c(t)$: cumulative number of intervals that have been started by time t.

$D_c(t)$: cumulative number of intervals that have been ended by time t.

$n(t)$: number of active intervals at time t.

The variables of the problem can be defined as

$d^l_p(t)$: cumulative number of passengers dispatched on line l by time t.

$a^l_b(t)$: cumulative number of vehicle trips dispatched on line l by time t.

$d^l_b(t)$: cumulative number of vehicle trips finished on line l by time t.

The size of the fleet, N is assumed bigger than $n(t)$ $\forall t$.

The total cost in this case has two components: the total delays suffered by the passengers and the operation cost of the vehicles. The delays can be estimated as the sum of the differences between $A_p^l(t)$ and $d_p^l(t)$. If an homogeneous value of time α and an operation cost per unit of trip time β are considered by the agency, then the total cost can be expressed as:

$$\sum_{l=1}^{L}\left[\int_{t=0}^{T}\alpha\left\{A_p^l(t)-d_p^l(t)\right\}dt+\beta R_l a_b^l(T)\right]$$

As in section 3.2 a feasible solution must satisfy the following sets of constraints:

$$d_b^l(t)=a_b^l(t-R_l), \qquad \forall l \in L, \forall t \in [1,T]$$

$$d_p^l(t)=\begin{cases} d_p^l(t-\epsilon)+min(K_l, A_p^l(t)-d_p^l(t-\epsilon)) & \text{if a vehicle departs on line } l \text{ at } t, \\ & \forall l \in L, \forall t \in [1,T] \\ d_p^l(t-\epsilon) & \text{otherwise.} \end{cases}$$

$$\sum_{l=1}^{L}\{a_b^l(t)-d_b^l(t)\} \leq n(t)=A_c(t)-D_c(t), \qquad \forall t \in [1,T]$$

In this section it is assumed that $a_b^l(t)$ is a continuous function instead of a step function. Thus, $a_b^l(t)$ is smoothed as a function going through the middle of its steps as is shown in figure 3.3. These curves overestimate the delay for some passengers and underestimate it for others. Over long periods of time the error incurred with this approach is small. According to this new function, the headway between vehicles dispatched on line l at t is given by the multiplicative inverse of the slope of $a_b^l(t)$, that is $[a_b^{l,}(t)]^{-1}$. The number of passengers arriving to the hub during the length of a headway can be estimated as $A_p^{l,}(t)/a_b^{l,}(t)$. Therefore, when vehicle capacity constraints are not met, the average number of passengers waiting at the hub, $q_p^l(t)$, is $A_p^{l,}(t)/2a_b^{l,}(t)$. In these periods, the passengers departure curve $d_p^l(t)$ can be estimated as

$$d_p^l(t) = A_p^l(t) - q_p^l(t) = A_p^l(t) - \frac{A_p^{l,}(t)}{2a_b^{l,}(t)}$$

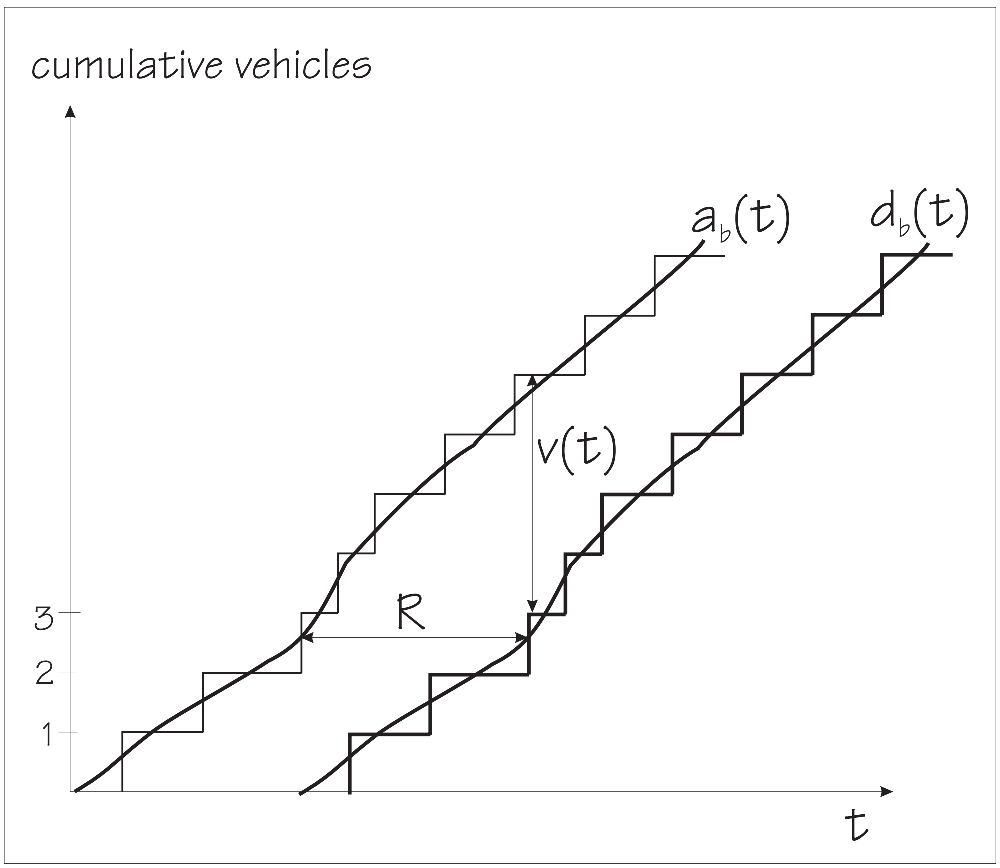

Figure 3.3: Smoothed $a_b^l(t)$ for line l

However, in periods when the number of passengers arriving during a headway exceeds the capacity of the vehicle, then a residual queue remains after the vehicle leaves. This residual queue is called $\tilde{q}_p^l(t)$. Therefore the queue of passengers at any time t, $q_p^l(t)$ is estimated as

$$q_p^l(t) = \tilde{q}_p^l(t) + \frac{A_p^{l'}(t)}{2a_b^{l'}(t)}$$

And the passengers departure curve $d_p^l(t)$ can be estimated more generally as

$$d_p^l(t) = A_p^l(t) - \frac{A_p^{l'}(t)}{2a_b^{l'}(t)} - \tilde{q}_p^l(t)$$

Figure 3.4 illustrates these two components of the total queue. In periods when $\tilde{q}_p^l(t)$ is not zero, $d_p^l(t)$ grows and recedes according with the total supply of passengers seats being offered, that is:

$$d_p^{l'}(t) = K_l a_b^{l'}(t)$$

Thus, in these periods, the rate of change of the queue of passengers at the hub is:

$$q_p^{l'}(t) = A_p^{l'}(t) - K_l a_b^{l'}(t)$$

and the residual queue can be estimated as[2]:

[2]This assumes that $\frac{d}{dt}\frac{A_p^{l'}(t)}{2a_b^{l'}(t)}$ can be neglected. This is correct if the functions involved change smoothly.

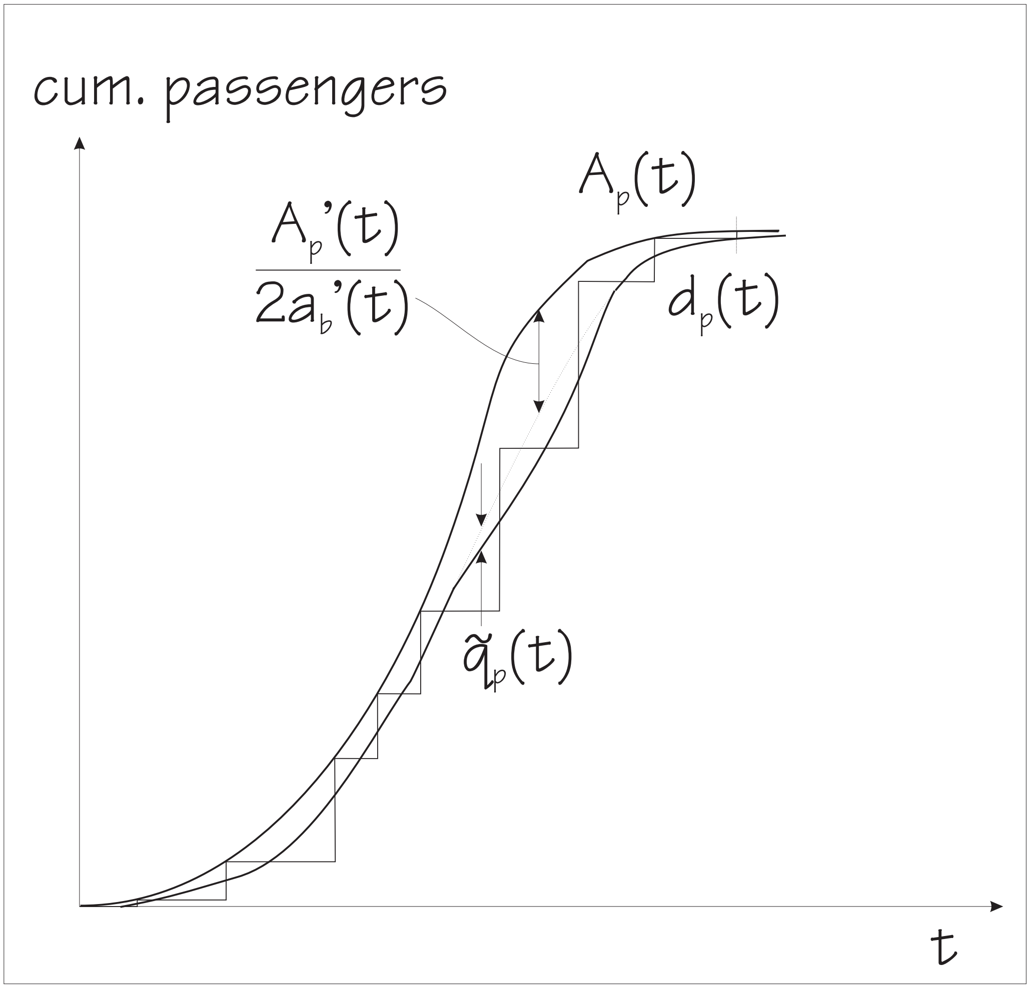

Figure 3.4: Two components of $q_p^l(t)$ for a single line case: a stationary and a residual queue

$$\tilde{q}_p^l(t + \Delta t) = max\{0, \tilde{q}_p^l(t) + [A_p^{l,}(t) - K_l a_b^{l,}(t)]\Delta t\}$$

Therefore, the residual queue can take one of two values:

$$\tilde{q}_p^l(t) = 0 \qquad \text{or} \qquad \tilde{q}_p^{l,}(t) = A_p^{l,}(t) - K_l a_b^{l,}(t)$$

This analysis, built on the work done by [25], allows one to state the following mathematical program for the scheduling problem:

$$min \sum_{l=1}^{L} \int_{t=0}^{T} \left[\alpha(\tilde{q}_p^l(t) + \frac{A_p^{l,}(t)}{2a_b^{l,}(t)}) + \beta R_l a_b^{l,}(t) \right] dt \tag{3.3a}$$

subject to

$$\tilde{q}_p^l(t) \geq 0 \qquad \forall t \in [1,T], \forall l \in L \tag{3.3b}$$

$$\tilde{q}_p^{l,}(t) + K_l a_b^{l,}(t) \geq A_p^{l,}(t) \qquad \forall t \in [1,T], \forall l \in L \tag{3.3c}$$

$$\sum_{l=1}^{L} [a_b^l(t) - d_b^l(t)] \leq n(t) \qquad \forall t \in [1,T] \tag{3.3d}$$

$$d_b^l(t) - a_b^l(t-R) = 0 \qquad \forall t \in [1,T], \forall l \in L \tag{3.3e}$$

$$\tilde{q}_p^l(0) = a_b^l(0) = d_b^l(0) = \tilde{q}_p^l(T) = 0 \qquad \forall l \in L \tag{3.3f}$$

Notice that this problem is convex (convex objective function and linear constraints). In the absence of constraints (3.3d) and (3.3e) the optimal strategy consists on dispatching vehicles at a rate proportional to the square root of the passenger arrival rate unless the capacity constraint is met, in which case vehicles should be dispatched as fast as possible (or equivalently as soon as they become full), see [25][3]. That is:

$$a_b^{l,}(t) = max \left\{ \frac{A_p^{l,}(t)}{K_l}, \sqrt{\frac{\alpha A_p^{l,}(t)}{2\beta R_l}} \right\} \tag{3.4}$$

[3] In this reference, the author assumes an infinite reservoir of vehicles in the trip's origin. Those vehicles do not return to the origin when the trip is finished

This conclusion can be obtained after realizing that the problem can be decomposed by time slices.

However, the timetable design problem is harder since constraints (3.3d) link the optimal strategy at time t with what has happened in the R_l previous units of time on each line. Indeed, some counterintuitive results arise from this property. For instance, intuitively, if drivers are available earlier, then vehicles should not be dispatched later. Although this might be true for most cases, it is not always true. For instance consider two scenarios with driver functions $n_i(t), i = 1, 2$ and their corresponding optimal dispatching rates $a_b^i(t), i = 1, 2$. The following example proves that:

$$n_1(t) \geq n_2(t) \forall t \nRightarrow a_b^1(t) \geq a_b^2(t) \forall t$$

Consider the following case, where ϵ is a small positive value:

$$A_p(t) = \begin{cases} t & 0 \leq t \leq 1, \\ 1 + \epsilon(t-1) & 1 \leq t \leq 2. \end{cases}$$

Then if $n_1(t) = 1, \forall t \in [0, 3]$, $K = 1$, $R = 1$ and $\beta = 0$ then the optimal strategy is to dispatch vehicles at $t = 1$ and $t = 2$. However, if the available drivers are instead:

$$n_2(t) = \begin{cases} 1 & \forall t \in [1-\epsilon, 2-\epsilon]U[2,3], \\ 0 & \text{elsewhere.} \end{cases}$$

then vehicles should be dispatched at $t = 1 - \epsilon$ and $t = 2$. Thus, $a_b^1(1-\epsilon) < a_b^2(1-\epsilon)$ although $n_1(t) \geq n_2(t) \forall t$.

In the next section, a Taylor polynomial expansion of first degree on $a_b^l(t)$ is explored. This approximation makes the problem decomposable by time slices and allows one to obtain a closed form function for the optimal $a_b^l(t)$.

First degree approximation on $a_b^l(t)$

In this section $a_b^l(t - R_l)$ is approximated as

$$a_b^l(t - R_l) \approx a_b^l(t) - R_l a_b^{l,}(t) \tag{3.5}$$

If (3.5) is replaced in (3.3e), and the resultant (3.3e) in (3.3d) the following constraint is obtained:

$$\sum_{l=1}^{L} R_l a_b^{l,}(t) \leq n(t) \tag{3.6}$$

Notice that now the problem can be decomposed by time slices. At each time slice dispatching rates and queue lengths must satisfy three constraints: $\tilde{q}_p^l(t) \geq 0$, $\tilde{q}_p^{l,}(t) + K_l a_b^{l,}(t) \geq A_p^{l,}(t)$ and $\sum_{l=1}^{L} R_l a_b^{l,}(t) \leq n(t)$.

As expected, the more drivers available the less likely the drivers constraint will become active. In the case of a single line (L=1), at any instant in time at most one of the two constraints (capacity and drivers) will be binding and therefore the optimal cost per unit time at t can be defined as:

$$\text{optimal cost per unit time} = \begin{cases} \frac{\alpha R A_p^{;}(t)}{2n(t)} + \beta n(t), & \text{if the drivers constraint binds at t} \\ \alpha\left(\tilde{q}_p(t) + \frac{K}{2}\right) + \frac{\beta R A_p^{;}(t)}{K}, & \text{if the capacity constraint binds at t} \\ \sqrt{2\alpha\beta R A_p^{;}(t)}, & \text{otherwise} \end{cases}$$

In the general case of many lines, the problem is more complicated. If bus capacities (K_l) are high enough so that buses can accommodate the demand at all stops, then the problem is simpler since then $q_p^l(t) = 0, \forall t$. Thus, the problem can be formulated as follows:

$$minL = \sum_{l=1}^{L} \int_{t=0}^{T} \left[\alpha(\frac{A_p^{l;}(t)}{2a_b^{l;}(t)}) + \beta R_l a_b^{l;}(t)\right] dt + \int_{t=0}^{T} \lambda_t \{\sum_{l=1}^{L} R_l a_b^{l;}(t) - n(t)\} dt \quad (3.7a)$$

subject to

$$a_b^{l;}(t) \geq 0 \qquad \forall t \in [1, T], l \in L \quad (3.7b)$$

$$\lambda_t \geq 0 \qquad \forall t \in [1, T] \quad (3.7c)$$

$$a_b^l(0) = 0 \qquad \forall l \in L. \quad (3.7d)$$

where the driver availability constraint (3.6) has been moved to the objective function as in standard Lagrangean relaxations. Since this problem is convex and can be decomposed by time slices, the optimality condition is:

$$\frac{\partial L}{\partial a_b^{l,}(t)} = 0 \Rightarrow \begin{cases} a_b^{l,}(t) = \sqrt{\frac{\alpha}{2}\frac{A_p^{l,}(t)}{(\beta R_l + \lambda_t R_l)}} \\ \lambda_t = \max\{0, \left[\frac{\alpha}{2}\frac{A_p^{j,}(t)}{[a_b^{j,}(t)]^2} - \beta R_j\right]\frac{1}{R_j}\} \end{cases} \quad \forall j = 1 \ldots L \tag{3.8}$$

so at any moment t when the drivers constraint is binding:

$$\sum_{l=1}^{L} R_l a_b^{l,}(t) = n(t) \Rightarrow \sum_{l=1}^{L} R_l \sqrt{\frac{\alpha}{2}\frac{A_p^{l,}(t)}{(\beta R_l + \frac{R_l}{R_j}\frac{\alpha}{2}\frac{A_p^{j,}(t)}{[a_b^{j,}(t)]^2} - \beta R_j \frac{R_l}{R_j})}} = n(t) \quad \forall j \in 1 \ldots L \tag{3.9}$$

which is equivalent to:

$$\sum_{l=1}^{L} R_l a_b^{j,}(t) \sqrt{\frac{R_j A_p^{l,}(t)}{R_l A_p^{j,}(t)}} = n(t) \qquad \forall j \in 1 \ldots L \tag{3.10}$$

which yields:

$$a_b^{j,}(t) = \frac{n(t)}{R_j}\frac{\sqrt{A_p^{j,}(t) R_j}}{\sum_{l=1}^{L}\sqrt{A_p^{l,}(t) R_l}} = \kappa n(t)\sqrt{\frac{A_p^{j,}(t)}{R_j}} \qquad \forall j \in 1 \ldots L \tag{3.11}$$

Notice that when the driver constraint is binding, the solution is not sensitive to the parameters α and β, only to passenger arrival rates and roundtrip durations. This could have been immediately observed from problem (3.7) since if drivers are binding then $\sum_{l=1}^{L} R_l a_b^{l,}(t)$ turns to be constant. And therefore the term in the objective function corresponding to the operational cost is constant as well. Equation (3.11) indicates that when drivers are a scarce resource, short and crowded lines should be priorized. Also notice that if all routes are equally long, then the optimal $a_b^{j,}(t)$ is

$$a_b^{j,}(t) = \frac{n(t)}{R} \frac{\sqrt{A_p^{j,}(t)}}{\sum_{l=1}^{L} \sqrt{A_p^{l,}(t)}} \qquad \forall j \in 1 \ldots L \quad (3.12)$$

This approach yields only an approximation of the optimal cost for problem (3.2). Since the solution of problem (3.2) provides a lower bound for problem (3.1), the continuum approximation cost should approximate the optimal cost for the timetable design problem on the low side.

Validation

To test the approximations of section 3.3.3 computational experiments are developed. The daily demand per line is assumed proportional to the one in figure 2.4b. The procedure consists of solving the timetable design problem using the closed formulas (3.11) and then using the numerical optimization approach to test the quality of the approximation. However, to solve the problem exactly, conventional integer programming tools are not enough since for medium size problems, the optimality gaps obtained in this way are too wide (up to 70%).

Thus, cutting planes specifically designed for model (3.2), and a polynomial-time separation algorithm to identify the most violated cut associated with each fractional solution are developed. These cutting planes when embedded in a branch-and-cut tree make problem (3.2) tractable. Interestingly, the cutting planes designed for this problem are also valid and new for the production lot sizing problem (see [2]) which has caught broad attention in the literature. All the information regarding these cuts and the separation algorithm is explained in chapter 4.2. The uninterested reader can skip it without missing the argument line of this dissertation.

Single line cases

The first set of experiments considers a system with just one line. Its roundtrip duration is assumed to be 1 hour. The size of the time step is set in 5 minutes so that a (19 hours) operating day consists of 235 time periods. The waiting time cost of a passenger is assumed to be \$10/hr while the operational cost, \$24/hr.

Initially a problem is created where drivers and capacity constraints does not bind. In this case, the continuous approximation based on [25] predicts that 78.3 buses would be dispatched over the whole day and that the total daily cost would be \$11, 269. The numerical optimization solution is very similar: 76 buses and a daily cost of \$11, 330. That is the approximated formula underestimates the total cost by just 0.5%.

Then, the same problem is solved with drivers heavily binding. In this case the approximation of equation (3.6) predicts a total cost of \$17, 605 and 79 buses. Figure 3.5a displays this solution along with the driver availability constraint and the unconstrained solution. Although the driver availability constraint reduces the number of vehicles operating considerably, the approximated solution is still sometimes over the constraint. The numerical optimization predicts 77 buses to be used at a cost of $18,146, that is, the cost is underestimated in less than 3%. Thus, insufficient drivers result in more delays for passengers while the number of buses dispatched (and therefore the operational costs) are kept rather constant. This solution is displayed in figure 3.5b along with the approximated solution. Notice that the first degree approximation follows reasonably well the optimal solution and therefore it seems to predict the optimal policy quite well.

Multiple line cases

Next, a 3-lines system is tested. In this case the operating day is discretized over 10-minute intervals yielding 114 time periods. As before, the operation cost of a vehicle is assumed to be 2.4 times more valuable than the waiting cost of a passenger during the same period. The demand for trips for each line is shown in figure 3.6. The number of drivers available at each moment is assumed given and its daily shape is chosen approximately proportional to the demands. As seen from the results, the availability of drivers is an active constraint during long periods during the day. The capacity of the vehicles is assumed not binding.

In the first example, the roundtrip duration of all lines is assumed to be one hour. The total cost predicted by the closed formula is $13,843.7. Then the problem is

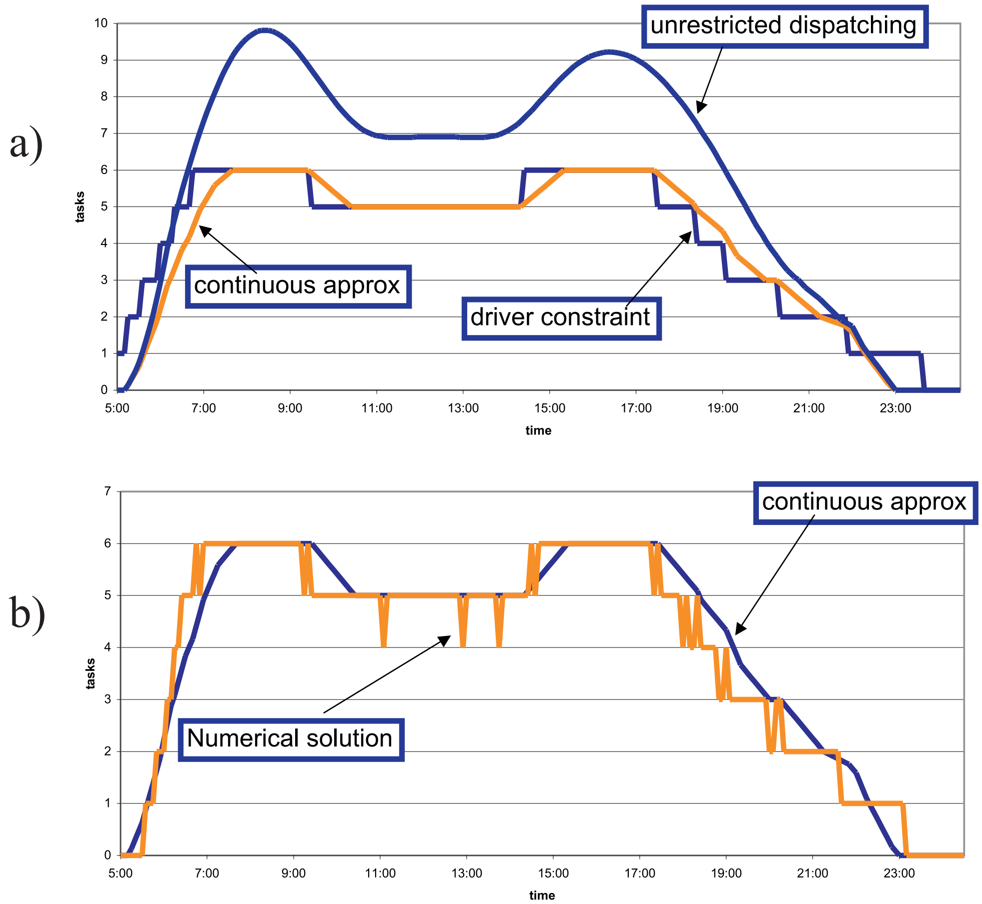

Figure 3.5: Solution for given driver schedule, a) approximated and unrestricted solutions, b) numerical and approximated solutions

solved to optimality using numerical optimization tools explained in chapter 4 yielding an optimal cost of \$14,280.5, thus the formula underpredicts the optimal cost in 3.06%. Figure 3.7 shows the number of drivers available, the number of drivers needed according with the optimal solution and the number of drivers needed according with the approximated procedure, along the day. Notice that the approximated procedure does not provide a feasible solution. However, a feasible solution with a

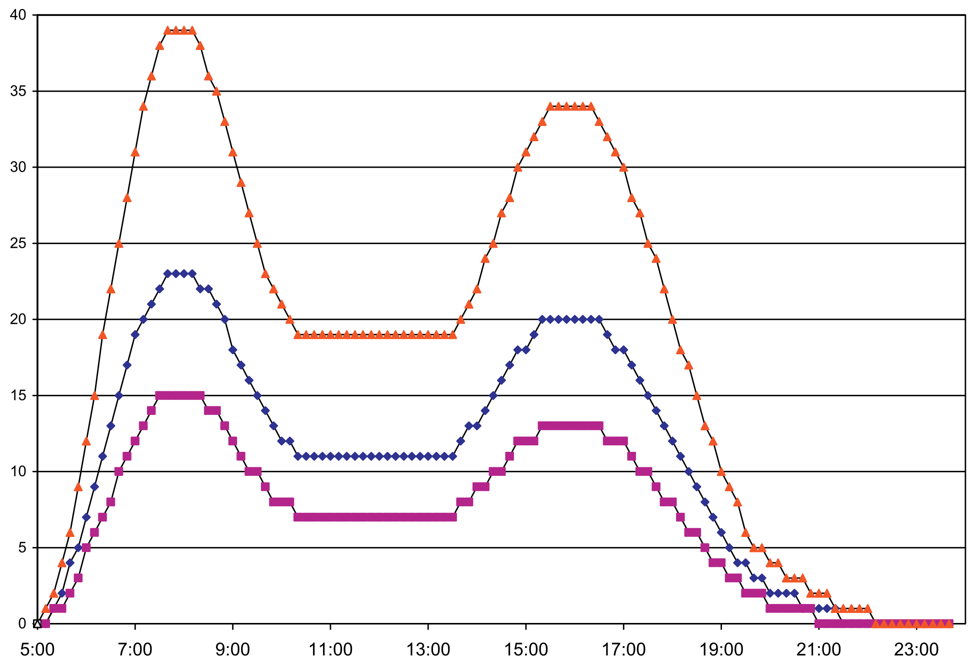

Figure 3.6: Daily demand for a three lines example

cost within 0.5% of the optimal cost can be obtained following a greedy heuristic.

In a second example, roundtrip durations are no longer constant along lines. Instead, the lines with high, medium and low demand have a roundtrip duration of 80, 60 and 40 minutes, respectively. The total cost predicted by the approximated formula is $14,010.8. The approximated problem is then solved to optimality yielding a cost of $14,551.0, thus the formula underpredicts the optimal cost in 3.71%.

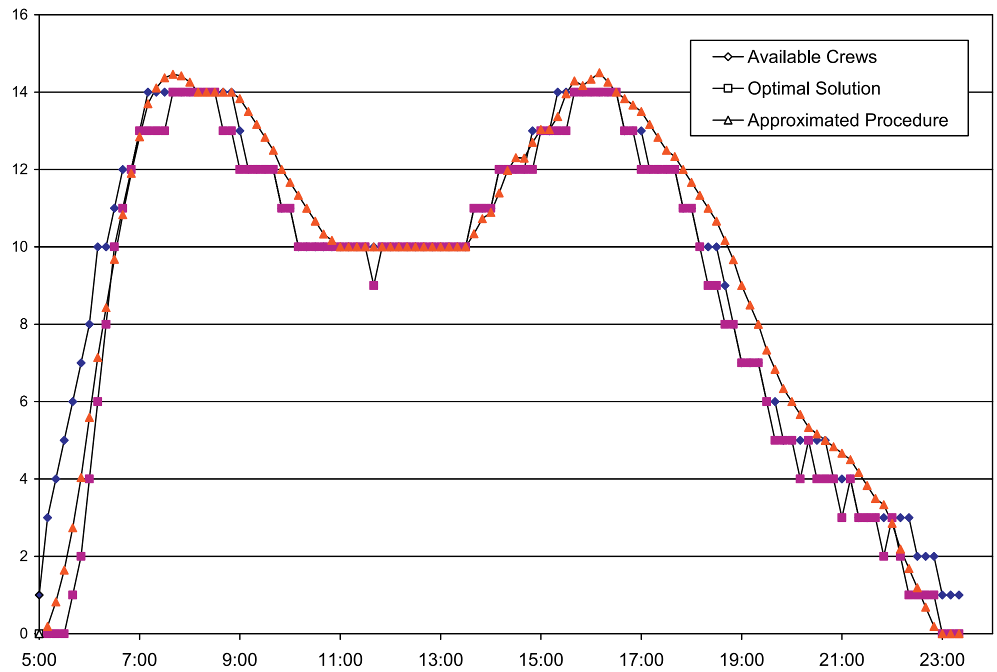

Figure 3.7: drivers available and required along the day for the optimal and approximated solution

3.4 The Stochastic Problem

3.4.1 Absenteeism

In this section absenteeism is considered. As in section 2.5, each day the agency will know exactly which drivers will come to work before operation begins. The agency tries to assign trips to the working drivers so that trippers and overtime pay are minimized.

Formulations (3.3) and (3.7) are no longer valid since the number of drivers actually working at every time t is a random variable ($n(t)$ is only the upper limit of that variable). As in section 2.5, it is assumed here that drivers are independent and identical, with daily show-up probability p. At every instant t the number of driver attending their jobs distributes binomial with mean $pn(t)$. The cumulative distribution function for this variable is called $\varphi_t(x)$.

Following the same strategy explained in section 2.5, if the agency lacks drivers to operate the system, it will hire overtime and trippers. These drivers are paid at a premium wage w_t. Thus, the objective function of this problem is reformulated as the expected cost.

$$\min L = \int_{t=0}^{T} \left[\sum_{l=1}^{L} \left\{ \alpha \frac{A_p^{l,}(t)}{2a_b^{l,}(t)} + \beta R_l a_b^{l,}(t) \right\} + w_t \int_{x=0}^{\sum_{l=1}^{L} R_l a_b^{l,}(t)} \left\{ \sum_{l=1}^{L} R_l a_b^{l,}(t) - x \right\} d\varphi_t(x) \right] dt \tag{3.13a}$$

subject to

$$a_b^{l,}(t) \geq 0 \qquad \forall t \in [1, T], \forall l \in L \tag{3.13b}$$

$$a_b^{l}(0) = 0 \qquad \forall l \in L. \tag{3.13c}$$

The first term in the objective function corresponds to the waiting time plus operational cost as in previous models. The last term corresponds to the expected payment for trippers.

Notice that the problem can still be decomposed in time slices. The derivative of the objective function with respect to $a_b^{m,}(t)$ is:

$$\frac{\partial L}{\partial a_b^{m,}(t)} = 0 \Rightarrow a_b^{m,}(t) = \sqrt{\frac{\alpha}{2} \frac{A_p^{m,}(t)}{(\beta R_m + R_m w_t \varphi_t(\sum_{l=1}^{L} R_l a_b^{l,}(t)))}} \tag{3.14}$$

Thus, the solution is very similar to the non-absenteeism problem. Here the Lagrangian multiplier λ_t takes the value $w_t\varphi_t(\sum_{l=1}^{L} R_l a_b^{l,}(t))$. Notice that if the system is highly overstaffed, $\varphi_t(\sum_{l=1}^{L} R_l a_b^{l,}(t)) \approx 0$ yielding the dispatching rate suggested by [25].

The solution for this problem could be easily achieved using the following algorithm:

Step 1: $n = n(t)$

Step 2: Compute $a_b^{m,}(t) = \sqrt{\frac{\alpha}{2}\frac{A_p^{m,}(t)}{(\beta + R_m w_t \varphi_t(n))}}, \forall m \in L$

Step 3: If $\sum_{l=1}^{L} R_l a_b^{l,}(t) < n$ then redefine n as $n - \delta$ where δ is a small value and go to step 2.

Step 4: Stop.

3.4.2 Stochastic Passenger Arrivals

Consider the case when the passenger demand per time slice is uncertain and a probability distribution is assumed instead. Notice that when bus capacity is never reached, passenger demand only appears in the objective function of problem (3.7). Since its effect is linear, the optimal solution for stochastic passenger arrivals can be obtained by replacing $A_p^{l,}(t)$ by $E[A_p^{l,}(t)]$ in the deterministic problem.

However, if the vehicles can reach capacity then the solution is not as straightforward.

3.5 A Joint-Design Example

To conclude this chapter, the example of section 3.3.3 is analyzed further to check if a better solution can be obtained by iterating with the shift filling problem. The optimal timetable aggregated across the three lines yielded by the timetable design problem is displayed in figure 3.8. Notice that it is considerably less smooth than those analyzed in previous examples. For the shift design problem the following combination menu are considered:

Combination 1 of table 2.1: Eight hours a day, five-days a week.

Combination 4 of table 2.2: Three days working ten hours and two days only five.

Combination 8: One split shift of four hours in the morning and four hours in the evening five days a week.

where the split shift combination is paid 50% more than the first two combinations. In this case the total cost improves only 0.01% after the shift filling problem is solved for the timetable of figure 3.8. The problem then converged since the timetable design problem constrained to the drivers availability yielded the same timetable as in the previous iteration.

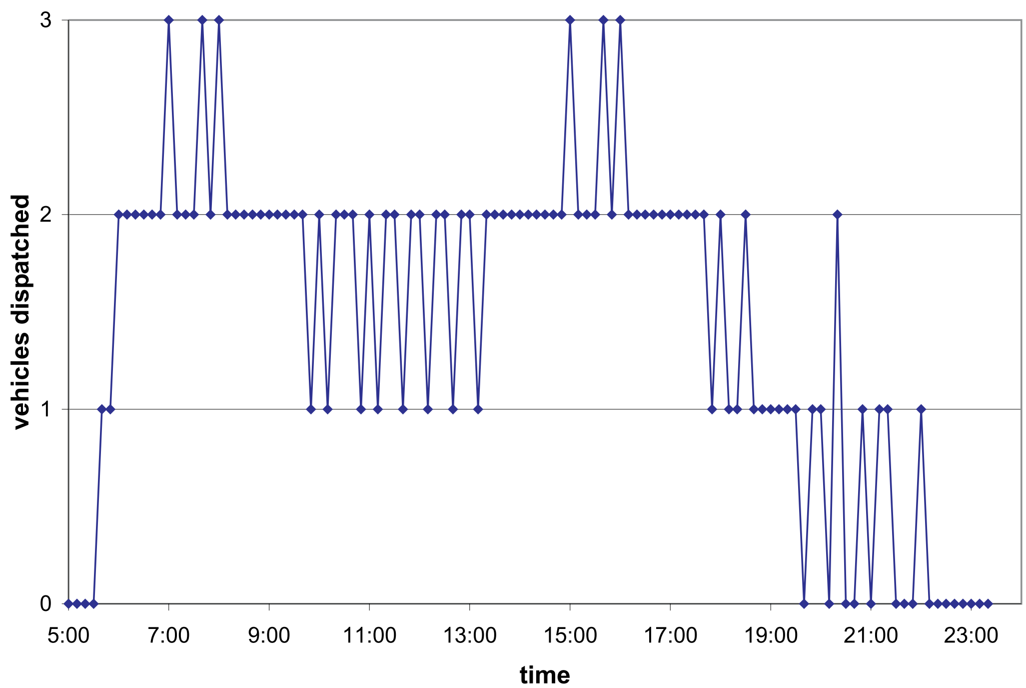

Figure 3.8: Aggregated timetable for the three-lines example

Chapter 4

Combinatorial Approach

This chapter explains a methodology used to solve the timetable design problem (3.2) exactly when no uncertainties are involved. To do this, cutting planes are designed, their validity is proven and a separation algorithm to determine the most violated cut is introduced. The complexity of the separation algorithm is shown to be polynomial.

For the reader not interested in combinatorial optimization tools, this chapter can be skipped without loss in the argument of this dissertation.

4.1 Analysis of the Problem

As before, let L be the number of lines and T be the number of time periods. The feasible set associated with problem (3.2) can be rewritten as the following sets of equations:

$$\sum_{l=1}^{L} \sum_{i=t-R_l}^{t} x_i^l \leq n_t \quad \forall t \in [1 \dots T] \tag{4.1}$$

$$y_t^l \leq x_t^l \quad \forall l \in [1, L], \forall t \in [1, T] \tag{4.2}$$

$$\sum_{v=1}^{t} y_v^l \leq u_t^l \quad \forall l \in [1, L], \forall t \in [1, T] \tag{4.3}$$

$$x_t^l \in \{0, 1\} \quad \forall l \in [1, L], \forall t \in [1, T] \tag{4.4}$$

$$y_t^l \geq 0 \quad \forall l \in [1, L], \forall t \in [1, T]. \tag{4.5}$$

where y_t^l now represents the number of passengers, in units of a vehicle capacity, that are loaded on line l at time t if a vehicle is dispatched. Thus, y_t^l takes values between 0 and 1. Accordingly, u_t^l now represents the cumulative demand by time t on line l in vehicle capacity units. By doing this linear transformation the vehicle capacities are normalized without loss of generality.

Notice that the inequalities in the polytope defined by:

$$P = conv\{x \in \{0, 1\}^{LxT} : x \text{ satisfying } (4.1 - 4.5)\}$$

have a subjacent totally unimodular matrix since all its elements are integer and all determinants of the submatrices are either $0, 1$ or -1. This means that if the right hand side of the inequalities are integer then all vertices of the polytope are integer as well. In this case the integrality constraints could be relaxed and a feasible solution would still be obtained. However, the extreme points of the polytope are not necessarily integer since the right hand side is not always integer.

Let's now focus on constraints (4.2) and (4.3). Consider the following single-lane set:

$$F = \left\{ x \in \{0,1\}^n, y \in \mathbb{R}_+^n : \begin{array}{r} y_1 \leq u_1 \\ y_1 + y_2 \leq u_2 \\ \vdots \\ y_1 + y_2 + \cdots + y_n \leq u_n \\ y_1 \leq x_1, \; \cdots, \; y_n \leq x_n \end{array} \right\}$$

Without loss of generality, it will be assumed that $0 \leq u_i - u_{i-1} \leq 1$ for all $i \in [1, n]$, where $u_0 = 0$. If the condition is not satisfied then the upper bound on t would be impossible to reach and it would be equivalent to assume in the formulation that the extra passengers (those that would be impossible to load at t) are instead arriving in the following time period. This should be repeated as preprocessing until the above condition is reached.

In the next section facet defining cutting planes for this polytope are developed. Their validity will be proved and the facet defining conditions identified.

4.2 Cutting Planes

Before introducing the general family of cutting planes, next section illustrates some geometric logic behind the cuts.

4.2.1 Illustration of valid cuts

Let's consider the case $n = 2$ were $x_1 = 1$:

$$F_2 = \left\{ x \in \{0,1\}^2, y \in \mathbb{R}^2_+ : \begin{array}{r} y_1 \leq u_1 \\ y_1 + y_2 \leq u_2 \\ y_1 \leq 1, y_2 \leq x_2 \end{array} \right\}$$

In this case figure 4.1a displays the set F_2 with the thickest lines and the convex hull of F_2 (from now on $Conv(F_2)$) with less thicker lines. Figure 4.1b shows the polyhedron defined by the linear relaxation of F_2. Notice that the two convex sets do not match and the latter includes an extreme fractional solution on x_2 highlighted with a circle. Therefore, the polyhedron obtained by relaxing the binary constraints does not correspond to $Conv(F_2)$. Consider the following plane that cuts this fractional solution without cutting any feasible solution for the original problem:

$$y_1 + y_2 - (u_2 - u_1)x_2 \leq u_1 \tag{4.6}$$

This tells us that if a vehicle is not dispatched on $t = 2$, then the upper bound of y_2 can be updated to u_1, otherwise the upper bound of y_2 must still be u_2. This constraint is therefore feasible and cuts the undesired fractional solution since it corresponds to the equation of the face of $Conv(F_2)$ in figure 4.1a that cuts the fractional solution. Also notice that by including these cuts in the formulation the upper bound constraints can be eliminated. It should be clear from the picture that

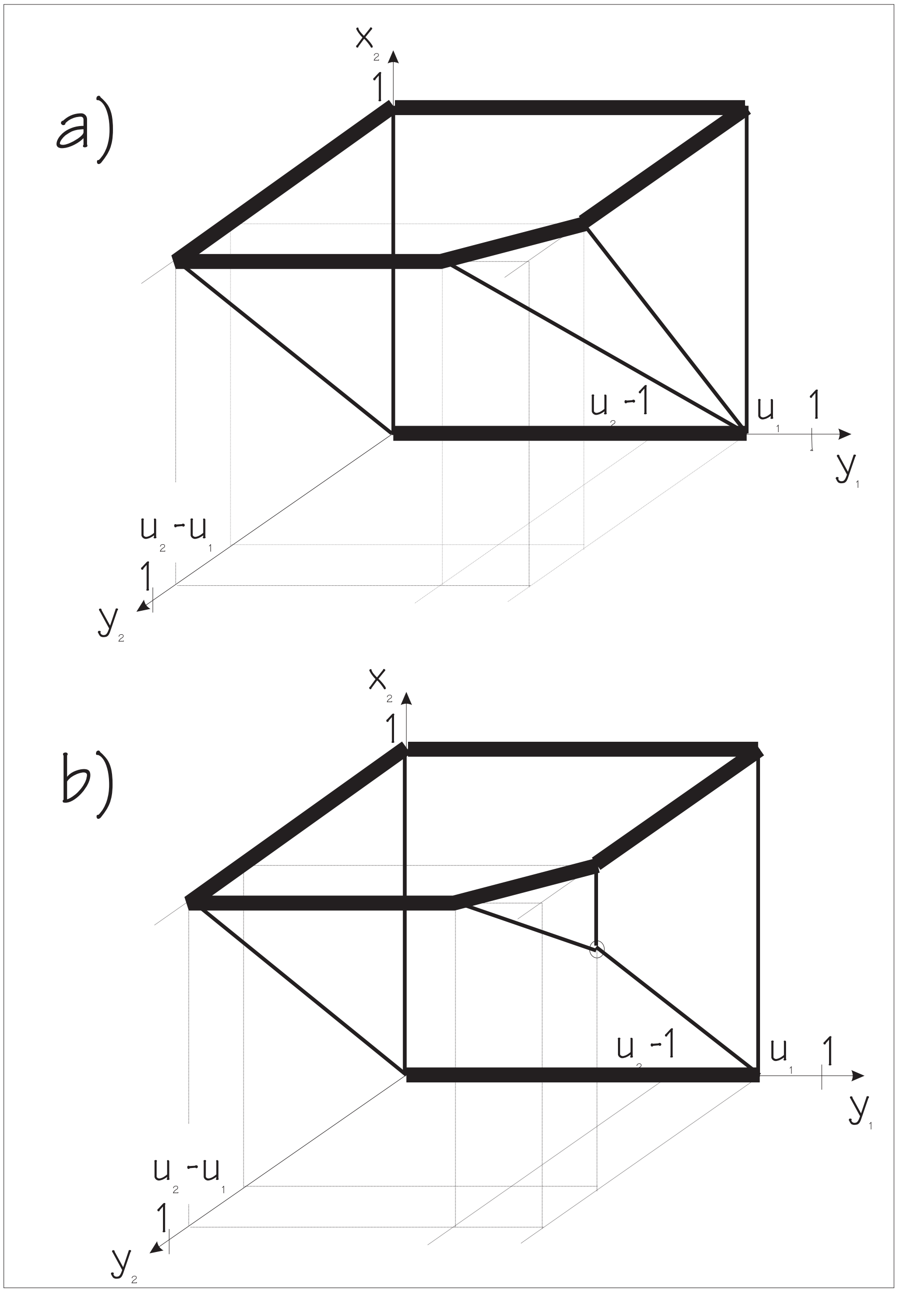

Figure 4.1: a) $Conv(F)$; b) polyhedron defined by the linear relaxation of F

if $u_2 - u_1 > 1$ then the cut is not a facet of the polyhedron. However, since it is assumed that $u_2 - u_1 < 1$ then the cut is (in general) a facet defining candidate.

The cut can be generalized with identical arguments for the relation between the set of variables $\{y_1, \ldots y_j, x_j\}, \forall j \in [1, n]$:

$$\sum_{k=1}^{j} y_k - (u_j - u_{j-1})x_j \leq u_{j-1} \tag{4.7}$$

This family of cuts can be generalized by lifting those variables not included in the inequalities. This procedure yields to a more general family of cuts which validity will be shown in the next section.

4.2.2 General family of cuts

Before introducing the general family of cuts, some definitions are needed. Consider a set $S = \{s_1, s_2, ..., s_p\} \subseteq [1, n]$ such that $s_i < s_{i+1}, \forall i \in [1, p-1]$. Let's also define $s_0 = 0$.

Then, note that if vehicles will be only dispatched during periods in S, then the maximum number of passengers that can be transported during those periods is:

$$\zeta(S) \equiv \max\{\sum_{i \in S} y_i : (x, y) \in F\} = \min_{0 \leq i \leq p} \{u_{s_i} + p - i\}. \tag{4.8}$$

$\zeta(S)$ will be called the *maximum output of S*.

Definition 1 The smallest minimizer *($q \in [0, p]$)* in (4.8) is called *bottleneck of S.*

Definition 2 For a given $i \in [1, p]$, we say that the *bottleneck of i,* denoted as q_i, is the bottleneck of the *set* $\{s_1, s_2, \ldots, s_{i-1}\}$.

Definition 3 $B_S \equiv \{q_i : i \in [1, p]\}$ is called the *bottleneck set of S*.

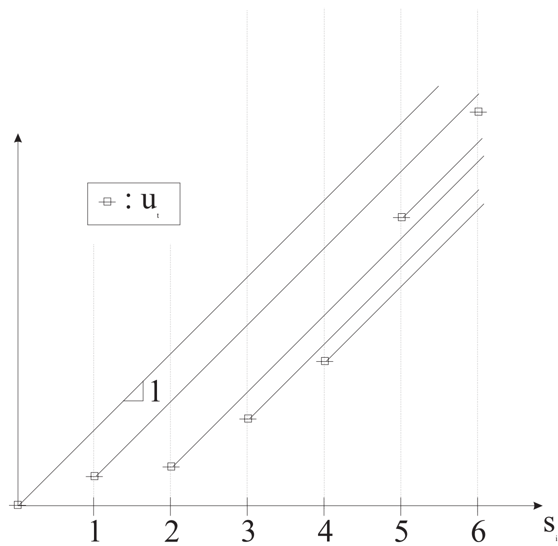

Figure 4.2: Function u_t defined for each element of S and the *maximum output* associated with each element

These definitions are illustrated with the following example. Consider a set S of 6 time periods ($p = 6$) and their cumulative passenger demands ($u_{s_i}, i \in [1, 6]$) as shown in figure 4.2; $u_{s_0} = 0$ is also included in the figure. Notice that from each point $(s_i, u_{s_i}), i \in [0, 6]$ a ray of slope 1 has been drawn. For each period s_i its ray represents the maximum number of passengers that could be loaded by each of the future periods if $\sum_{j=1}^{i} y_{s_j} = u_{s_i}$ and if future passenger demands were never met. Then, the bottleneck of S would be given by the period associated with the lowest ray passing through s_p. In the example, the bottleneck of S is 4. Similarly, the bottlenecks of $1, 2, 3, 4, 5$ and 6 are $0, 1, 2, 3, 4$ and 4, respectively. Therefore, the bottleneck set of S is in this case $B = \{0, 1, 2, 3, 4\}$.

Observation 4 *If p is the bottleneck of $S = \{s_1, s_2, \ldots, s_p\}$, then the following are true:*

1. $\zeta(S \setminus \{s_k\}) = \min\{u_{s_p}, u_{s_{q_k}} + p - q_k - 1\}$.
2. $\zeta(S \cup \{k\}) = \begin{cases} u_{s_p} & \text{if } k < s_p, \\ \min\{u_{s_p} + 1, u_k\} & \text{if } k > s_p. \end{cases}$

To illustrate this observations consider the set $S = \{1, 2, 3, 4\}$ from figure 4.2 whose bottleneck is 4. Then, if s_1 or s_2 are eliminated from S then the maximum output of the new set S is still u_{s_4}. However, if s_3 or s_4 are eliminated then the maximum output would become $u_{s_2} + 1$ or u_{s_3} in each respective case.

On the other hand, if an element $k < s_4$ is added to S then the maximum output of S would still be u_{s_4}. However, if $k > s_4$ then the maximum output will be given by the minimum of $u_{s_4} + 1$ and u_k.

Definition 5 *Given $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n]$, let's call*

$\lambda_i := u_{s_p} - u_{s_{q_i}} - (p - q_i - 1)$ for $i \in [1, p]$. In the diagram (figure 4.2), if s_i would be taken out of S, λ_i would be seen as 1 minus the vertical difference at $t = s_p$ between the position of the ray coming from s_{q_i} and u_{s_p}.

Now, S is called a bottleneck cover if $\lambda_i < 1$ for some $i \in [1, p]$.

Observation 6 *For any $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n]$ the following statements are true:*

1. $\lambda_i \leq \lambda_j$ *for $i, j \in [1, p]$ with $i \leq j$,*

2. *$\lambda_i = \lambda_j$ if and only if $q_i = q_j$ for $i, j \in [1, p]$,*

3. *$\lambda_i < \lambda_{i+1}$ if and only if $q_{i+1} = i$ for $i \in [1, p-1]$.*

4. *$\lambda_p > 0$ if and only if s_p is the bottleneck of S.*

For a bottleneck cover S, we define the *bottleneck cover inequality* as

$$\sum_{i=1}^{p} \min\{1, \lambda_i^+\}(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \leq u_{s_p}. \tag{4.9}$$

where $\lambda_i^+ = max\{\lambda_i, 0\}$.

Proposition 7 *Let LF denote the linear programming relaxation of F.*

(i) Every fractional extreme point of LF is defined by a bottleneck cover.

(ii) Bottleneck cover inequalities (4.9) cut–off all fractional extreme points LF.

Proof.

(i) Let (x, y) be an extreme point of LF.

Observe that $x_i \in \{1, y_i\}$, and $0 < x_i < 1$ implies that

$0 < y_i < 1$ and $y_i = u_i - \sum_{j=1}^{i-1} y_j$.

Now let $S = \{s_1, s_2, \ldots, s_p\} = \{i \in [1, n] : y_i > 0\}$ such that $0 < x_{s_p} < 1$

and let $k = \max\{i \in [0, p-1] : x_{s_i} < 1\}$.

Because either $k = 0$, or $k \geq 1$ and $y_{s_k} = u_{s_k} - \sum_{i=1}^{k-1} y_{s_i}$, we have

$y_{s_p} = u_{s_p} - u_{s_k} - (p - 1 - k)$. Since $y_{s_p} < 1$, we have $u_{s_p} < u_{s_k} + p - k$, and since (x, y) is feasible,

we have $u_{s_k} + j - k \leq u_{s_j}$ for all $j \in [k+1, p-1]$.

Then, by induction on k, S is a bottleneck cover and k is the bottleneck of p.

Hence, $\{i \in [1, p] : 0 < x_{s_i} < 1\}$ is precisely the bottleneck set of S, B_S.

(ii)

Since $y_{s_p} > 0$, we have $\lambda_p = u_{s_k} + p - k - u_{s_p} < 1$.

Then inequality (4.9)

defined by $S = \{i \in [1, n] : y_i > 0\}$ cuts off

(x, y) as $\sum_{i=1}^{p} y_{s_i} = u_{s_p}$ and $0 < x_{s_p} < 1$.

Proposition 8 *The family of inequalities* (4.9) *is valid for* F*.*

Proof. Consider some $(x, y) \in F$ and let $\{z_1, z_2, \ldots, z_\ell\} = \{1 \leq i \leq p : x_i = 0\}$, indexed in increasing order of i. The result holds trivially if $\lambda_{z_i} \leq 0$ for all $i \in [1, \ell]$. Now let $k = \min\{j \in [1, \ell] : \lambda_{z_j} > 0\}$. Then

$$\sum_{i=1}^{p} \min\{1, \lambda_i^+\}(1 - x_{s_i}) = \sum_{j=k}^{\ell} \min\{1, \lambda_{z_j}^+\} \tag{4.10}$$

$$\lambda_{z_k} + \ell - k \leq \max_{j=1,\ldots,\ell} (\ell - j + \lambda_{z_j})^+. \tag{4.11}$$

since $\min\{1, \lambda_{z_j}^+\} = 1$ for $j > k$ and k is one element in $[1 \ldots \ell]$. We also have

$$\sum_{i=1}^{p} y_{s_i} \leq \min\{u_{s_p}, \min_{j=1,\ldots,\ell}\{u_{s_{q_{z_j}}} + \big[(p - q_{z_j}) - (\ell - j + 1)\big]\}\} \tag{4.12}$$

$$= u_{s_p} - \max_{j=1,\ldots,\ell}\{(u_{s_p} - u_{s_{q_{z_j}}} - p + q_{z_j} + \ell - j + 1)^+\} \tag{4.13}$$

$$= u_{s_p} - \max_{j=1,\ldots,\ell}\{(\ell - j + \lambda_{z_j})^+\}. \tag{4.14}$$

The first inequality is true since both elements in the right hand side are upper bounds on the left hand side sum. The latter two equalities come from accommodating the elements and using the definition of λ_i.

Adding (4.11) and (4.14), shows that inequality (4.9) is satisfied by (x, y).

Proposition 9 *The inequality* (4.9) *defines a facet of* $conv(F)$ *if and only if:*

1. $\lambda_p < 1$, $max_{i \in [1,p]}\{\lambda_i\} > 0$,

2. $[1, s_{q_r}] \subset S$, *where* $r = min\{i \in [1, p] : \lambda_i < 1\}$, *and*

3. $K = \{k \in [1, n] \setminus S : u_{q_r} \leq u_k \leq u_t\} = \emptyset$, *where* $t = max\{i \in [q_r, r-1] : u_{s_{q_r}} + i - q_r = u_{s_i}\}$.

Proof. *Necessity.*

1. If $\lambda_p \geq 1$ means by definition that $u_{s_{q_p}} + p - q_p \leq u_{s_p}$, therefore some $s_{q_p} < s_p$ is the bottleneck of $S = \{s_1, s_2, \ldots, s_p\}$. So $\lambda_i = \lambda_p \geq 1$ for all $i = q_p + 1, \ldots, p$. Additionally, since $\lambda_p \geq 1$, $p - q_p \leq u_{s_p} - u_{s_{q_p}}$. Therefore we conclude that the inequality with S equal to $\{s_1, s_2, \ldots, s_{q_p}\}$ dominates (4.9), since $\sum_{i=q_p+1}^{p}(\min\{1, \lambda_i^+\}(1 - x_{s_i}) + y_{s_i}) = \sum_{i=q_p+1}^{p}((1 - x_{s_i}) + y_{s_i}) \leq p - q_p \leq u_{s_p} - u_{s_{q_p}}$.

2. If $\max_{i \in [1,p]}\{\lambda_i\} \leq 0$, then

   $y_1 + \cdots + y_{s_p} \leq u_{s_p}$ dominates (4.9).

3. Consider $k \in [1, s_{q_r}] \setminus S$. Since q_r is the bottleneck of $[1, s_{q_r}]$, by Observation 2, augmenting S with k, does not change λ_i for $i \in [q_r + 1, p]$ and does not increase λ_i for $i \in [1, q_r]$. Since the coefficients of all x_{s_i} with $i \in [1, r-1]$ are zero, the inequality with S equal to $\{s_1, \ldots, s_p\} \cup \{k\}$ dominates (4.9).

4. Finally, note that if $t > q_r$, then t is an alternative minimizer in (4.8), achieving the value $\zeta(\{s_1, \ldots, s_{r-1}\})$. Thus S can be augmented with $k \in K$ without changing λ_i for $i \in [r, p]$ to get a stronger inequality than (4.9).

Sufficiency. It is easy to check that the following $2n$ points (x^k, y^k) are affinely independent points of the face generated by (4.9). For $k \in [1, p]$ such that $0 < \lambda_k < 1$, let $y_{s_i}^{s_k} > 0$ and $x_{s_i}^k = 1$ for $i \in [1, q_k]$ such that $\sum_{i=1}^{q_k} y_{s_i} = u_{s_{q_k}}$, $y_{s_k}^{s_k} = 0$ and $x_{s_k}^{s_k} = 0$, and $y_{s_i}^{s_k} = 1$ and $x_{s_i}^{s_k} = 1$ for $i \in [q_k + 1, p] \setminus \{k\}$; and $y_i^{s_k} = x_i^{s_k} = 0$ for $i \in [1, n] \setminus S$; and let $y_{s_i}^{s_k} > 0$ and $x_{s_i}^k = 1$ for $i \in [1, q_k]$ such that $\sum_{i=1}^{q_k} y_{s_i} = u_{s_{q_k}}$, $y_{s_k}^{s_k} = \lambda_k$ and $x_{s_k}^{s_k} = 1$, and $y_{s_i}^{s_k} = 1$ and $x_{s_i}^{s_k} = 1$ for $i \in [q_k + 1, p] \setminus \{k\}$; and $y_i^{s_k} = x_i^{s_k} = 0$ for $i \in [1, n] \setminus S$.

For $k \in [1, p]$ such that $\lambda_k \leq 0$, let $y_{s_i}^{s_k} > 0$ and $x_{s_i}^k = 1$ for $i \in [1, q_k]$ such that $\sum_{i=1}^{q_k} y_{s_i} = u_{s_{q_k}}$, $y_{s_k}^{s_k} = 0$ and $x_{s_k}^{s_k} \in \{0, 1\}$, and $y_{s_i}^{s_k} > 0$ and $x_{s_i}^{s_k} = 1$ for $i \in [q_k + 1, p] \setminus \{k\}$ such that $\sum_{i=q_k+1}^{p} y_{s_i} = u_{s_p} - u_{s_{q_k}}$ and $y_i^{s_k} = x_i^{s_k} = 0$ for $i \in [1, n] \setminus S$.

Let $\bar{y} = \sum_{i=1}^{q_r} \epsilon_i e_{s_i} + \sum_{i=q_r+1}^{p} e_{s_i} - e_{s_r}$ and $\bar{x} = \sum_{i=1}^{p} e_{s_i} - e_{s_r}$ such that $1 \geq \epsilon_i > 0$ and $\sum_{i=1}^{q_r} \epsilon_i = u_{q_r}$. Since $\bar{y}$ has a slack of λ_r for constraint $y_1 + \cdots + y_{s_p} \leq u_{s_p}$ and s_p is the bottleneck of S, $\bar{y}$ has a positive slack for all constraints $y_1 + \cdots + y_i \leq u_i$ for $i > s_t$.

The remaining $2(n-p)$ points are described as follows: $(\bar{y}, \bar{x}+e_k)$ and $(\bar{y}+\epsilon e_k, \bar{x}+e_k)$ for $k \in [s_t, s_p] \setminus S$ with small $\epsilon > 0$; and (y^{s_p}, e_k) and $(y^{s_p} + \min\{1, u_{s_p+1} - u_{s_p}\}e_k, e_k)$ for $k \in [s_p + 1, n]$.

Lifting bottleneck covers

It was proven that the cuts given by inequality (4.9) are valid and the conditions under which they are facet defining were shown. In this section, the bottleneck cover inequalities (4.9) are generalized by introducing pairs of variables (x_i, y_i), $i \subseteq [s_p + 1, n]$ into the inequality. Let $T \subset [1, n]$ and consider the restriction $F_T = \{(x, y) \in F : x_i = y_i = 0 \text{ for } i \in T\}$ of F and a facet–defining bottleneck inequality

$$\sum_{i=1}^{p} \lambda_i^+(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} \leq u_{s_p} \tag{4.15}$$

defined by some $S = \{s_1, s_2, \ldots, s_p\} \subseteq [1, n] \setminus T$ for $conv(F_T)$. A new inequality of the following form will be derived

$$\sum_{i=1}^{p} \lambda_i^+(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} + \sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq u_{s_p} \tag{4.16}$$

starting from (4.15).

Let $F(u)$ be the set F as a function of the righthand side vector u. $F_T(u)$ is defined similarly. Let $\Phi : \mathbb{R}^n_+ \mapsto \mathbb{R} \cup \{+\infty\}$ be defined as

$$\Phi(v) = u_{s_p} - \max\left\{\sum_{i=1}^{p}(\lambda_i^+(1 - x_{s_i}) + y_{s_i}) : (x, y) \in F_T(u - v)\right\}.$$

By definition of Φ, inequality (4.16) is valid for $F(u)$ if and only if

$$\sum_{i \in T}(\pi_i x_i + \mu_i y_i) \leq \Phi(\sum_{i \in T} y_i g_i) \tag{4.17}$$

for all $(x, y) \in F(u)$, where $g_i = \sum_{k=i}^{n} e_k$, $i \in [1, n]$ and e_k is the kth unit vector.

Rather than characterizing $\Phi(\sum_{i \in T} y_i g_i)$ for all $(x, y) \in F(u)$, a lower bound on $\Phi(ag_\ell)$ for $0 \leq a \leq 1$ and $\ell \in T$ will be described, which suffices to prove the validity of the inequalities introduced in this section. For some $\ell \in T$ let $P_{\Phi(ag_\ell)}$ denote the problem of computing $\Phi(ag_\ell)$. Since (x_i, y_i), $i \in [1, n] \backslash S$ do not appear in inequality (4.15), they may be ignored when computing $\Phi(ag_\ell)$.

$$
\begin{aligned}
\Phi(ag_\ell) = u_{s_p} - \max \sum_{i=1}^{p} (\lambda_i^+(1-x_{s_i}) + y_{s_i}) & \\
\text{s.t.:} \qquad y_{s_1} &\le u_{s_1} \\
y_{s_1} + y_{s_2} &\le u_{s_2} \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_{s_p} \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_\ell - a \\
&\vdots \\
y_{s_1} + y_{s_2} + \cdots + y_{s_p} &\le u_n - a \\
y_{s_i} \le x_{s_i},\ y_{s_i} \in \mathbb{R}_+,\ x_{s_i} \in \{0,1\} \quad i \in [1,p]. &
\end{aligned} \tag{4.18}
$$

Claim 10 *Problem $P_{\Phi(ag_\ell)}$ has an optimal solution (x,y) such that*

1. *$x_{s_i} = 1$ for all $i \in [1,p]$ with $\lambda_i \le 0$,*

2. *$1 > y_{s_k} > \lambda_k^+$ for at most one $k \in [1,p]$, and*

3. *$y_{s_i} \in \{0,1\}$ for all $i \in [1,p] \setminus \{k\}$*

   with $\lambda_i > 0$,

4. *if $\sum_{i=1}^{h} y_{s_i} = u_{s_h}$ for $h \in [1,p]$,*

   then $y_{s_i} > 0$ for all $i \in [1,h]$ with $\lambda_i > 0$.

Proof.

Part 1 is immediate. For part 2, suppose $1 > y_{s_i} > \lambda_i^+$ and $1 > y_{s_j} > \lambda_j^+$ for $i < j$. Increasing y_{s_j} and decreasing y_{s_i} by the same amount sufficiently we either satisfy $y_{s_j} = 1$ or $y_{s_i} = \lambda_i^+$ and do not increase the objective function value. To see part 3, observe that if $y_{s_i} \le \lambda_i < 1$, the objective function improves by $\lambda_i - y_{s_i}$ by setting $y_{s_i} = x_{s_i} = 0$.

Part 4 is a consequence of feasibility. By definition of a bottleneck, $u_h + p - h \ge u_{q_h} + \sum_{i=q_h+1}^{p} = u_{s_p} + \lambda_h$. Since for any $j \in [1, h]$, we also have $u_{q_j} + p - q_j = u_{s_p} + 1 - \lambda_j$, it follows that $u_h \ge u_{q_j} + h - q_j - (\lambda_h - \lambda_j)$ for $j \le h$. Therefore, if $\lambda_j > 0$, since $\lambda_h \le 1$ ((4.15) is facet–defining), we have $u_h > u_{q_j} + h - q_j - 1$ and $y_{s_j} = 0$ it contradicts with $\sum_{i=1}^{h} y_{s_i} = u_{s_h}$.

Since (4.15) is facet–defining for $conv(F_T)$, we have $\Phi(0) = 0$, and consequently $\Phi(ag_\ell) \ge 0$ for $a \ge 0$. From Observation 4, we see that $\Phi(0) = 0$ is achieved by (x, y) such that $x_{s_i} = 1$ for all $i \in [1, p] \setminus \{k\}$ and $x_{s_k} = 0$ for any $k \in [1, p]$ with $\lambda_k > 0$, and $\sum_{i=1}^{p} y_{s_i} = u_{s_p} - \lambda_k$. Since (4.18) has a slack of λ_k for this solution for all $a \le \delta_\ell := u_\ell - u_{s_p}$, we have $\Phi(ag_\ell) = 0$ for $0 \le a \le \delta_\ell + \lambda_p$, since $p = \text{argmax}_{i \in [1,p]}\{\lambda_i\}$.

Then for $a = \delta_\ell + \lambda_p$ there exists an optimal $(\bar{x}, \bar{y})$ such that $\sum_{i=1}^{q_p} \bar{y}_{s_i} = u_{s_{q_p}}$, $\bar{y}_{s_i} = 1$ for all $i \in [q_p + 1, p - 1]$, and $\bar{y}_{s_p} = 0$; so that $\sum_{i=1}^{p} \bar{y}_{s_i} = u_{s_p} - \lambda_h$. Since constraint (4.18) is tight at this point, given that $\bar{x}_{s_h} = 0$, increasing a beyond $\delta_\ell + \lambda_h$ requires reduction in some $\bar{y}_{s_i}$, $i \in [1, p - 1]$, which will increase $\Phi(ag_\ell)$ at the same rate.

Since $1-\lambda_p = \min_{i\in[1,p]}\{1-\lambda_i\}$, the value of any $y_{s_d} = 1$ can be reduced by $1-\lambda_d$ and then set $y_{s_d} = x_{s_d} = 0$ without changing the objective function, and introduce a slack of $a_{s_d} - 1 + \lambda_d$ to constraint (4.18) again. Thus $\Phi(ag_\ell)$ will increase in the interval $a = [\delta_\ell + \lambda_p, \delta_\ell + \lambda_d]$ at a rate of 1. Since this interval goes beyond $a = 1$, $\Phi(ag_\ell)$ has been characterized over the interval of interest.

Once $\Phi(ag_\ell)$ has been determined, valid coefficients (π_ℓ, μ_ℓ) are obtained for (x_ℓ, y_ℓ) by ensuring that

$$
\begin{aligned}
h(a) = \max\ & \pi_\ell x_\ell + \mu_\ell y_\ell \\
\text{s.t. } & y_\ell = a \\
& 0 \le y_\ell \le a_\ell x_\ell,\ x_\ell \in \{0,1\}
\end{aligned}
$$

is no more than $\Phi(ag_\ell)$ as suggested in Gu et al [21] in this case. It is seen above that

Thus π_ℓ and μ_ℓ are the intercept and slope of $\Phi_\ell(a)$. That is:

$$\pi_\ell = -(\delta_\ell + \lambda_p) \text{ and } \mu_\ell = 1$$

Thus the validity of the following new family of valid inequalities has been shown:

$$\sum_{i=1}^{p} \lambda_i^+(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} + \sum_{i\in T}(-(u_i - u_{s_p} + \lambda_p)x_i + y_i) \le u_{s_p} \tag{4.19}$$

However, if $u_i - u_{s_p} + \lambda_p > 1$, the cut is dominated by the inequality (4.16) in which:

$\pi_\ell = -1$ and $\mu_\ell = 1$

Thus, the new inequality can be rewritten as:

$$\sum_{i=1}^{p} \lambda_i^+(1 - x_{s_i}) + \sum_{i=1}^{p} y_{s_i} + \sum_{i \in T}(-\min(1, u_i - u_{s_p} + \lambda_p)x_i + y_i) \leq u_{s_p} \qquad (4.20)$$

Although this family of cuts is exponential in size, a polynomial separation algorithm to identify the most violated cut from this family has been developed. The algorithm is explained in the next section.

4.3 Separation Algorithm

The family of inequalities (4.20) is exponential since the number of sets S grow exponentially with n. However, in this section, an algorithm to identify the set S yielding the most violated cut is developed. In order to do this, given a point (x, y), for each $s_p \in [1, n]$, a directed network is defined on which a longest path corresponds to an inequality with the largest left hand side value for (x, y).

Consider a network $N = (V, A)$, where each vertex in V is a triple (t, b, r) such that $t \in [0, s_p]$, $b \in [0, t-1]$, and $r \in [1, s_p - t]$ if $t \in [1, s_p - 1]$, $r = 0$ if $t = s_p$, and r is undefined if $t = 0$. For a vertex (t, b, r), t denotes an element that may possibly be included in S, b denotes a possible bottleneck for t, and r denotes $|\{s_i \in S : s_i > t\}|$. Each arc $((t_i, b_i, r_i), (t_k, b_k, r_k))$, or simply (i, k), in A satisfies

$t_i < t_k$, $r_i = r_k + 1$, and $b_k = t_i$ or $b_k = b_{t_i}$.

Thus N is an acyclic, directed network with the source vertex $(0, 0, -)$ and s_p sink vertices $\{(s_p, 0, 0), (s_p, 1, 0), \cdots, (s_p, s_p - 1, 0)\}$. The set of vertices on a path from the source vertex to one of the sink vertices represents S and an arc (ik) on such a path denotes that t_i and t_k are two consecutive elements in S. Observe that bottleneck of t_k is either t_i or the bottleneck of t_i; therefore, the network can be easily built with a forward pass.

Given a point (x, y) to separate with inequalities (4.20), the length of arc $(i, k) \in A$ equals the contribution of (x_k, y_k) to the left hand side of the inequality. In order to do that the length of (ik) is defined as $c_{ik} + \bar{c}_{ik}$ where

$$c_{ik} = \begin{cases} y_{t_k} + \min\{1, (u_{s_p} - r_k)^+\}(1 - x_{t_k}) & \text{if } t_i = 0 \\ y_{t_k} + \min\{1, (u_{s_p} - u_{b_i} - (\mathtt{p[i]} - \mathtt{p[d_k]} + r_k))^+\}(1 - x_{t_k}) & \text{if } t_i \geq 1, \end{cases}$$

$$\bar{c}_{ik} = \begin{cases} 0 & \text{if } t_k \neq s_p \\ \sum_{i \in s_{p+1}}^{n} y_i + (-(u_i - u_{s_p} + \min\{1, (u_{s_p} - r_k)^+\})x_i) & \text{if } t_i = 0 \\ \sum_{i \in s_{p+1}}^{n} y_i + (-(u_i - u_{s_p} + & \\ \qquad + \min\{1, (u_{s_p} - u_{b_i} - (\mathtt{p[i]} - \mathtt{p[d_k]} + r_k))^+\})x_i) & \text{if } t_i \geq 1 \end{cases}$$

where $\mathtt{p[i]}$ is the number of vertices in a longest path from the source vertex $(0, 0, -)$ to vertex i. Since the longest path algorithm on an acyclic directed network

[1] proceeds in topological ordering of the nodes, $\mathtt{p[i]} - \mathtt{p[b_k]}$ is determined before the arc (i, k) is used in the algorithm. Hence c_{ik} is computed when it is needed within the longest path algorithm. Given a longest path with arc (ik), since $\mathtt{p[i]}$ refers to the position of t_i in S, $p = |S|$, and $r_k = |\{s_i \in S : s_i > t_k\}|$, it is clear that $\mathtt{p[i]} - \mathtt{p[d_k]} + r_k = p - b_{p[k]} - 1$. Hence the length of a longest source to sink path equals the maximum left hand side value for any inequality (4.20) that has s_p as the maximum element in S.

The network has $O(n^4)$ arcs. Then separation problem of inequalities (4.20) is solved in $O(n^5)$ by running the linear–time longest path algorithm for each $s_p \in [1, n]$.

Unfortunately, although this algorithm's complexity is polynomial, it takes too long for all practical purposes to run. It is likely that the complexity of the separation algorithm can be reduced by a careful calculation. However computations showed that simple separation heuristics are sufficient to close almost all of the integrality gap at the root node. Thus, during this dissertation (unless noted) this separation algorithm is not used. Instead, a greedy heuristic is used to determine a finite collection of sets S in order to accelerate the time needed to solve the problems. This heuristic is discussed in Appendix A.

4.4 Computational Tests

In this section some computational experience on using the inequalities introduced in Section 4.2.2 for the deterministic timetable design problems is described. The inequalities are used as cutting planes on a branch–and–cut algorithm. To highlight the effect of the inequalities, the driver availability constraint of the timetable design problem was relaxed, that means that in these problems an unconstrained number of drivers was assumed. All experiments are done on a 2GHz Intel Pentium4/Linux workstation with 1GB RAM using the callable libraries of CPLEX[1] Version 8.1 Beta with one hour time limit.

The problems had the following characteristics:

1. The number of periods was $20, 50, 100$ or 200.
2. The number of lines was $5, 20$ or 50.
3. The demand on each time period was uniformly distributed $[0, 20]$.
4. The roundtrip time of every line was 1 hour.
5. The capacity of each vehicle was 150.
6. The operational cost of a vehicle was $120/hr
7. The waiting time of passengers was valued as $30/hr.

Each of the 12 problems was solved in two different ways:

1. With default CPLEX.
2. Using CPEX after turning off all its default cuts and preprocessing, but adding the lifted bottleneck cover cuts of section (4.2.2).

[1]CPLEX is a trademark of ILOG, Inc.

Tables 4.1, 4.2 and 4.3 display the results of these tests. They show how advantageous these cuts are for solving the timetable design problem since in these cases gaps are reduced dramatically and problems that otherwise look very hard, become solvable within one hour (with the exception of L=50, T=200 which after one hour the root node problem could not be finished).

These tables report the percentage integrality gap of the formulation before cuts are added ($\texttt{initgap} = 100 \times (\texttt{initub} - \texttt{initlb})/\texttt{initub}$), the percentage integrality gap after adding cuts before branching ($\texttt{rootgap} = 100 \times (\texttt{rootub} - \texttt{rootlb})/(\texttt{rootub})$), and the percentage gap between the lowest upper bound and the highest lower bound on the optimal value at termination ($\texttt{endgap} = 100 \times (\texttt{bestub} - \texttt{bestlb})/\texttt{bestub}$), where:

1. initlb: the objective function value of the initial LP relaxation,
2. initub: the first feasible solution found,
3. rootlb: LP relaxation after all cuts are added before branching,
4. rootub: best feasible solution found before branching,
5. bestlb: the lowest lower bound for all unexplored nodes of the search tree,
6. bestub: the best feasible solution.

The tables also include the the number of cuts added in the search tree (`cuts`), the number of nodes explored (`nodes`) and the elapsed CPU time in seconds (`time`).

T	initgap	rootgap	endgap	time	cuts	nodes
20	69.67	12.46	0.00	0.07	31	60
	93.37	0.00	0.00	0.04	68	0
50	87.71	46.94	27.57	3600	174	4205941
	93.26	0.05	0.00	1.94	348	3
100	91.01	51.66	24.27	3600	320	1439700
	93.35	0.05	0.00	15.35	741	3
200	92.19	70.26	52.10	3600	389	582326
	93.35	0.02	0.00	103.26	1597	2

Table 4.1: 5 lines

In these cases, the cuts provide a big improvement in all the problems. The root gaps are reduced dramatically and problems that look very hard are solved optimally within an hour. Notice that the biggest of the problems in the set has $10,000$ binary variables.

T	initgap	rootgap	endgap	time	cuts	nodes
20	68.48	8.92	0.00	31.30	131	54831
	93.10	3.11	0.00	0.22	275	1
50	87.78	59.38	37.37	3600	359	1373249
	93.37	2.56	0.00	14.81	1300	7
100	91.06	78.00	63.40	3600	378	477227
	93.35	0.03	0.00	130.33	3000	77
200	92.17	85.93	75.99	3600	380	146798
	93.27	0.02	0.00	908.94	6663	145

Table 4.2: 20 lines

T	initgap	rootgap	endgap	time	cuts	nodes
20	68.02	13.84	3.66	3600	267	3127495
	93.35	2.48	0.00	0.72	671	2
50	87.67	74.60	58.15	3600	351	632161
	93.35	1.86	0.00	61.23	3178	69
100	90.91	85.23	74.85	3600	352	184311
	93.22	4.54	0.00	664.34	7773	890
200	99.20	89.63	81.98	3600	352	44498
	93.30	4.96	0.00	8080.36	17079	7641

Table 4.3: 50 lines

Chapter 5

Conclusions

This chapter summarizes the results of this dissertation, discusses implementation issues and suggests ideas for future research.

5.1 Summary of Results

This research argues that driver contracts in the transit industry should be more flexible due to the double-peaked passenger demand curve. If instead of working 40 hours a week, 8 hours a day, drivers could choose among a set of daily work schedules still totalling 40 hours a week with two days off (heterogeneous contracts) without split shifts, then transit agencies could reduce their operational costs by about 10%. This reduction comes from savings on expensive split shifts and part time contracts, and fewer idle driver hours. In addition, these contracts could be designed to fit driver needs better than current ones. Evidence in [8] and gathered informally in this research show that most people prefer heterogeneous contracts like those described in chapter 2. Therefore, adopting these heterogeneous contracts might benefit both agencies and drivers.

Estimating the benefits of these new contracts required a planning tool that could predict the long term operation cost for different labor contracts. This operational cost must be estimated under an uncertain daily operation due to driver absenteeism and passenger demand. This research has provided such a tool.

This transit planning tool has been developed in three steps: optimizing labor contracts given the timetables, optimizing the timetables given the labor contracts, and jointly optimizing labor contracts and timetables. In the first step, timetables are assumed to be given and optimal labor contracts are determined. In this step the agency must balance the trade-off between hiring too many extra drivers and paying too much in overtime and trippers. Currently agencies do not seem to balance them adequately. A function that optimizes the number of drivers to hire given the timetables, absenteeism rate, and wages for regular drivers and trippers is proposed. With this function the impact of changes in parameters on the total cost can be quantified.

The second step looks at the complementary problem: contracts are given but timetables need to be determined. This dissertation approaches this problem in two different ways: using continuum approximation, and numeric optimization. With the continuum approximation a function describing the optimal dispatching rate as a function of roundtrip durations and passenger arrival rates has been developed. This function suggests that in the presence of insufficient number of drivers, priority should be given to short and crowded lines. The approach was extended to incorporate uncertainties. The numeric optimization approach was to solve the integer programming problem directly. Since the problem proved to be intractable for medium size instances, cutting planes were designed, their validity was proved, and facet defining conditions were identified. These cuts allowed large instances of the problem to be solved within an hour (up to 10,000 integer variable problems were tested). That is fast and precise enough that an agency could use it to adjust schedules if for a short period of time (few weeks or months) the system were to change (e.g. in passenger demand, in workforce, in the network).

The final step consisted of designing labor contracts and timetables simultaneously. This dissertation explored an iterative approach between the two problems explained above. When a flexible combination menu is considered, this problem can be solved with only one iteration.

5.2 Implementation

If heterogeneous combinations were to be implemented the first step is to understand drivers needs and diversity. The agency must ensure that the contracts offered meet those needs. This requires that the agency builds combinations that some drivers are willing to work and that the set of contracts to be offered captures the number of drivers willing to work each of those combinations. This dissertation has provided limited guidance for this procedure. Heterogeneous combinations can be implemented progressively, offering an increasing set of them each time the contracts are renewed. However, most of the savings can be gained offering few (attractive) heterogeneous contracts since the marginal benefit of introducing them should be non-increasing.

Heterogeneous combinations may face resistance from drivers and unions. Fewer drivers could be required to make the same number of trips and those working might have fewer weekly idle hours. However, these drawbacks could be mitigated if the agency were to extend the service with the same number of drivers and offer higher salaries for working heterogeneous combinations.

The planning procedure developed may also face driver resistance if it reduces the number of overtime hours. Many drivers expect to improve their salaries through overtime. However, the suggested procedure still relies on overtime and trippers to deal with absenteeism. Therefore, regular drivers should be the first ones to be offered tripper work.

All drivers have been assumed to be unionized and paid according with a very simple wage structure. However, the wage structure might be more complicated preventing the simpler analysis developed here.

This dissertation has also suggested a new procedure for assigning drivers to trips. However implementing it may need some adaptation. In this procedure contracts only identify the start and end time of each shift interval. The assignment to trips is done at the beginning of each day depending on which drivers show-up for work. This assignment assumes that all drivers can work any line. Additionally, this assignment also determines which drivers should arrive earlier or leave later than scheduled, and which trips should be assigned to trippers.

Developing a driver-trip assignment software should be simple. The heuristic explained in Appendix A provides a starting point that could easily be improved. Such a software could incorporate additional options to improve the applicability of its output (e.g., driver experience might be a requisite to work some trips, some drivers may not be capable to work any line, etc.). Adding this type of constraint would be simple, and since multiple optimal solutions will (most likely) exist for the driver-trip assignment, this might not increase the operational cost. This research assumes that drivers are flexible (i.e., any driver can drive any line). However, as in most systems, some flexibility should provide most of the benefit.

Prior knowledge of which drivers will come to work is crucial in this assignment procedure (as it is for transit agencies, currently). Therefore, drivers should be encouraged to announce an eventual absence as soon as possible. The strategy to determine how many drivers to hire relies on an accurate estimation of the absenteeism rate. This rate could be estimated based on historical data. The planning tools developed in this research allow the agency to determine the benefit of encouraging higher attendance among drivers[1].

[1] Notice that drivers improving their attendance unexpectedly might increase cost while the

The findings of this research are illustrated with a double-peaked passenger demand curve based on ACT data. However, the main assumption on this research is not the shape of this demand profile but its predictability. Therefore, these results are also applicable for systems facing very different demands.

5.3 Future Research

A natural extension of this work is considering more realistic network structures. However, transit systems usually have few garages where lines start and end. Therefore the system might be modelled as a set of hub and spoke structures like the one studied here. The resulting problem would need to incorporate penalties for moving drivers among hubs.

Additionally, fleet size has been omitted from the planning process. In the single hub case, peak frequency determines the size of the fleet. In multiple-hub networks vehicle synchronization between hubs would be needed. Incorporating this cost in the timetable design problem should be a future task.

For simplicity, operational uncertainties have not been considered in this dissertation. However, trip times are rarely constant across days [2] and vehicles sometimes fail, affecting the smooth operation of the system. Thus, a more robust design should

---

contracts are in place since the agency will appear to be overstaffed. Better attendances are beneficial for the agency only if they can be predicted

[2]Notice that roundtrip durations of a line (predictably) changing along the day can be easily incorporated in the methodology presented here.

account for these uncertainties.

Also, this research assumes that at the end of each day all drivers inform the agency if they will come to work on the next day. Although this assumption allows a fair comparison between the statu-quo and the proposed methodology, it is not very realistic. If show-up decision were to be taken in the last moment, and late show-up was also considered, then the driver-trip assignment would be much more complicated. This problem is left as future research.

Regarding the timetable design problem, the agency must balance passengers waiting time and operational costs. This political issue is often addressed by giving waiting times a monetary value (α). Since this is a political decision, it would be interesting to identify the impact of this parameter on the output (costs, frequencies, etc.) of the model.

A critical issue in this dissertation is identifying driver preferences recognizing their diversity and creating heterogeneous contracts accordingly. This research has only identified this diversity and described some tentative combinations but most of the work has not been done yet.

Finally, the planning approach developed for transit systems can also be studied for other systems also facing predictable peaked demands like delivery systems, call centers and emergency hospitals. However, these industries might have different characteristics or face different constraints preventing a straightforward application. Regarding package delivery systems, the routes are flexible, and packages are often segregated into regular and priority delivery. Additionally, the demand might not be as peaked, and delays might be considerably longer while still acceptable. However, the methodology described in this dissertation could be of value for different systems.

Bibliography

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs NJ, 1993.

[2] A. Atamtürk and J.C. Muñoz. A Study of the Lot-Sizing Polytope. Submitted to *Mathematical Programming*, 2002.

[3] H.S. Baker and O. Schueftan. Study of Operator absenteeism and workers' compensation trends in the urban mass transportation industry. Prepared for the U.S. Department of Transportation under contract #UMTA-PA-0050-80. Peat, Marwick, Mitchell & Co., Washington D.C.

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

[5] M.O. Ball and H. Benoit-Thompson. A lagrangian relaxation based heuristic for the urban transit crew scheduling problem, in *Computer-Aided Transit Scheduling*, J.R. Daduna and A. Wren (eds.), Springer-Verlag, Berlin. pp 54-67, 1988.

[6] Caprara, Toth, Vigo, Fischetti. Modeling and solving the crew rostering problem. *Operations Research*, 46: 820-830, 1998.

[7] A. Ceder. Personal communication, 2001.

[8] R.M. Coleman. *The 24 hour business. Maximizing productivity through the round-the-clock operations*, Amacom, New York, 1995.

[9] C.F. Daganzo. *Logistics Systems Analysis.* Springer, Berlin, 1999.

[10] C.F. Daganzo and G.F. Newell. Configuration of physical distribution networks. *Networks*, 16: 113-132, 1986.

[11] Day and Ryan. Flight attendant rostering for short-haul airline operations. *Operations Research*, 45: 649-661, 1997.

[12] G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis and D. Villneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems, in *Fleet Management and Logistics*, T.G. Crainic, G. Laporte (Eds.), Kluwer Academic, Norwell MA, 1998.

[13] M. Desrochers and F. Soumis. Bus crew scheduling by column generation. *Transportation Science*, 23: 1-13, 1989.

[14] J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis. Time constrained routing and scheduling, in *Handbooks in Operations Research and Management Sci-*

ence 8: Network routing, M.O. Ball, T.L. Magnanti, C.L.Monma and G.L. Nemhauser (eds.), North-Holland, Amsterdam, 35-139, 1995.

[15] S. Eilon, C.D.T. Watson-Gandy and N. Christofides. *Distribution Management.* Griffin, London, 1970.

[16] J.C. Falkner and D.M. Ryan. EXPRESS: Set partitioning for bus crew scheduling in Christchurch, in *Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop*, M. Desrochers and J.M. Rousseau (eds.), Springer, Berlin, 359-378, 1992.

[17] M. Fischetti, S. Martello and P. Toth. The fixed job schedule problem with working time constraints. *Operations Research*, 37: 395-403, 1989.

[18] R. Freling, A.P.M. Wagelmans and J.M.P. Paixão. An overview of models and techniques for integrating vehicle and crew scheduling. *Computer-Aided Transit Scheduling*, N.H.M. Wilson (ed.), Springer, Berlin. pp 441-460, 1999.

[19] M. Gamache, F. Soumis, G. Marquis and J. Desrosiers. A column generation approach for large-scale air crew rostering problems. *Operations Research*, 47: 247-263, 1999.

[20] A.M. Geoffrion. The purpose of mathematical programming is insight, not numbers. *Interfaces*, 7: 81-92, 1976.

[21] Z. Gu, G.L. Nemhauser and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4:109-129, 2000.

[22] A. Langevin, P. Mbaraga and J.F. Campbell. Continuous approximation models in freight distribution: an overview. *Transportation Research B*, 30B: 163-188, 1996.

[23] A. Mingozzi, M.A. Boschetti, S. Ricciardelli, L. Bianco. A set partitioning approach to the crew scheduling problem. *Operations Research*, 47: 873-888, 1999.

[24] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*, Wilsey-Interscience, New York, 1999

[25] G.F. Newell. Dispatching policies for a transportation route. *Trans Sci* 5: 91-105, 1971.

[26] G.F. Newell. Scheduling, location, transportation, and continuum mechanics: some simple approximations to optimization problems. *SIAM, Journal of Applied Mathematics*, 25:346-360, 1973

[27] A.R. Odoni, J.M. Rousseau and N.H.M. Wilson. Models in urban and air transportation, in *Handbooks in Operations Research and Management Science 6: Operations Research and the Public Sector*, S.M. Pollock, M.H. Rothkopf and A. Barnett (eds.), North-Holland, Amsterdam, 129-150, 1994.

[28] I. Patrikalakis and D. Xerocostas. A new decomposition scheme of the urban public transport scheduling problem, in *Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop*, M. Desrochers and J.M. Rousseau (eds.), Springer, Berlin, 407-425, 1992.

[29] J.M. Rousseau, R. Lessard and J.Y. Blais. Enhancement to the HASTUS crew scheduling algorithm, in *Computer Scheduling of Public Transport 2*, J.M. Rousseau (ed.), Elsevier, Amsterdam, 1985.

[30] F. Shepardson and R.E. Marsten. A lagrangean relaxation algorithm for the two duty period scheduling problem. *Management Science*, 26: 274-281, 1980.

[31] Y. Shiftan and N.H.M. Wilson. Absence, overtime, and reliability relationships in transit workforce planning. *Transportation Research A*, 28A: 245-258, 1994.

[32] K.R. Smilowitz and C.F. Daganzo. Cost Modeling and Design Techniques for Integrated Package Distribution Systems. Forthcoming in Transportation Science, 2003.

[33] P.H. Vance, C. Barnhardt, G.L.Nemhauser. Airline crew scheduling: a new formulation and decomposition algorithm. *Operations Research*, 45: 188-200, 1997.

[34] J. Walker. Personal communication, 2002.

[35] N.H.M. Wilson. Personal communication, 2002.

[36] A. Wren, B.M. Smith and A.J. Miller. Complementary approaches to crew scheduling, in *Computer Scheduling of Public Transport 2*, J.M. Rousseau (ed.), Elsevier, Amsterdam, 1985.

Appendix A

Upper Bound for the Filling Design Problem

In this section an heuristic that finds a feasible solution for the shift filling problem is presented. The inputs of the heuristic are a list of intervals and a set of trips. The heuristic assigns trips to intervals and determines if new intervals need to be considered, overtime hours need to be scheduled and/or trippers need to be hired.

Let's call $x_f(t)$ the number of intervals of type f to be started at time t, and $z(t, \ell)$ the number of trips to depart from the hub at time t on line ℓ. Let U be the total number of intervals, and $u = 0$. The output of this heuristic will be a list b of length U in which each element $b(u)$ will contain the set of trips assigned to interval u.

Consider the following heuristic:

Step 1. Initialization: order the intervals in F in decreasing interval type length. Let $b(u) = , \forall u \in [1, U]$. Let $f = 1$, $t = 1$, $u = 0$.

Step 2: Repeat substep 2 until $t = T$ and $u = U$. As a result of this step, each interval in the solution of the simplified model is assigned to a set of trips. Also, a set of trips not allocated to any shift are left in $z(t, \ell)$.

Substep 2. $u = u + 1$. If $x_f(t) > 0$ then the interval associated to $b(u)$ will be of length f. Assign trips to $b(u)$ in the following way: let $\tau = t$; repeat until $\tau > t + f$: if $z(\tau, \ell) > 0$ for a given ℓ and $\tau + r_\ell > t + f$ (the trip can be finished within the interval) do $z(t, \ell) = z(t, \ell) - 1$ (trips assigned are erased from $z(t, \ell)$) and $\tau = \tau + r_\ell$, else $\tau = \tau + 1$. If $x_f(t) = 0$ then do $t = t + 1$ or $f = f + 1$ if $t = T$.

Step 3. At this step $z(t, \ell)$ represents all trips for each line not assigned to intervals (elements in b). Call the total number of not assigned trips Z. Sort them by starting time in descending order and call the new list q where $q(i), i \in [1..Z]$ is the starting time of trip i in the list. Let $q(Z + 1) = 0$. Keep the roundtrip duration of each of these trips in a separate list.

Step 4. In this step, as many trips in $q(i)$ as possible are inserted into the elements of b. Let $i = 1$. While $q(i) > 0$ do substep 4.

Substep 4. $x = q(i)$. See if x can be accommodated in one of the elements of b, say $b(u)$, without reallocating other trips in the element. If yes, add the trip to the element of $b(u)$, and make $q(j) = q(j + 1), \forall j \in [i, Z]$. If x can not be inserted in any element of b, try to exchange trip x with some other trip y in the assignments such that y starts earlier than x and such that the new assignment (after exchanging x and y is still feasible). If more than one trip fits these conditions choose the one that starts the latest. If no trip y exists, then $i = i + 1$. If a trip y do exist, then try to insert it in an element of b without reallocating trips. If y can be inserted, do it; and make $q(j) = q(j + 1), \forall j \in [i, Z]$, if it can't be inserted, place y in q so that the decreasing order is kept.

Step 5. In this step, the the remaining trips on q need to be accommodated using extra shifts. Two cases are distinguished: the deterministic case ($p = 1$), and the stochastic case ($p < 1$). For the deterministic case, problem (2.5) can be run again using the trips in q as the input. Using the solution of that problem as an inputr, this heuristic can be run again to allocate the trips to the new set of intervals.

In the stochastic case, look if any trip in q can be done partially by an interval in b. If yes, assign the trip to the interval and consider the non-covered part of the trip as overtime. Once no more overtime opportunities are left, all trips are covered with trippers.

Appendix B

Alternative Objective Function for (2.14)

The goal of problem (2.14) is to find a set of intervals that will match the target curve as close as possible. The active intervals at any moment can be above or below the target curve. In this chapter, the cost of deviating from the target curve is estimated for small deviations using a second degree polynomial approximation.

If events are modelled over a continuous time reference, then the shift filling goal on the simplified formulation is to minimize the following objective function:

$$\min z = \int_{t=0}^{T} \left[wpn(t) + w_t \int_{x=-\infty}^{v(t)} \phi(x; n(t)p, n(t)p(1-p))(v(t)-x)dx \right] dt \quad \text{(B.1)}$$

Notice that in this expression, the only variable is the function n(t) representing the number of active intervals at time t. Let's call the integrand of the objective function $f(n(t))$. As was mentioned, $z^* = f(n^*(t))$ provides a lower bound in the optimal cost. However, if the target curve $n^*(t)$ can not be achieved (due to the rigidity of the combination menu), the lower bound may be improved if the cost of deviating from the target curve is identified. Thus, imagine that $n^*(t)$ is perturbed with a function $\epsilon(t)$. What would be the objective cost for $n(t) = n^*(t)+\epsilon(t)$? If $\epsilon(t)$ is small $\forall t$, the objective function (B.1) evaluated for this curve can be estimated using a second degree Taylor approximation as:

$$E[cost] = \int_{t=0}^{T} f(n^*(t)+\epsilon(t))dt \simeq \int_{t=0}^{T} \left[f(n^*(t)) + f'(n^*(t))\epsilon(t) + \frac{f''(n^*(t))\epsilon(t)^2}{2} \right] dt$$

However, from optimality conditions on $n^*(t)$, $f'(n^*(t)) = 0$. Thus:

$$E[cost] \simeq z* + \int_{t=0}^{T} \left[\frac{f''(n^*(t))\epsilon(t)^2}{2} \right] dt$$

that is the extra cost increases with the square of the size of the perturbation, no matter its sign. Let's look at $f''(n^*(t))$ after dropping the argument t of $n(t)$ and $v(t)$ for simplicity.

$$f''(n) = \frac{d}{dn} \left[\frac{d}{dn} \left[wnp + w_t \Psi(v; np, np(1-p)) \right] \right] \simeq w_t p^2 \phi(v; np, np(1-p))$$

In this last step the approximation from (2.10) is used. So the extra cost at time t of having $n(t)$ drivers instead of $n^*(t)$ is:

$$w_t p^2 \phi(v; np, np(1-p)) \frac{[n(t) - n^*(t)]^2}{2} = \frac{w_t p^2 e^{-\frac{(v-n^*(t)p)^2}{n^*(t)p(1-p)}}}{2\sqrt{2\pi p(1-p)}} \frac{[n(t) - n^*(t)]^2}{\sqrt{n^*(t)}}$$

However, optimality conditions on $n^*(t)$ (equation (2.11)) imply that:

$$z_{w/w_t} = \frac{v - n^*(t)p}{\sqrt{n^*(t)p(1-p)}}$$

Where z_{w/w_t} represents the $100\frac{w}{w_t}$ percentile on a cumulative Normal(0,1) distribution. Thus, the extra cost over the whole time period would be:

$$\frac{w_t p^2 e^{-(z_{w/w_t})^2}}{2\sqrt{2\pi p(1-p)}} \int_{t=0}^{T} \frac{[n(t) - n^*(t)]^2}{\sqrt{n^*(t)}} dt \tag{B.2}$$

This function provides the extra cost due to not matching the target curve perfectly if the deviation from the target is small.