Universitat Politècnica de Catalunya
Department of Statistics and Operations Research

Scheduling Policies for Multi-Period Services

PhD Candidate:
Alma Cristina Núñez del Toro

Thesis Advisor:
Elena Férnandez Aréizaga, PhD

November 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

To Marusa.

Acknowledgements

"If you want to go fast, go alone. If you want to go far, go together."
African proverb.

Along the development of this thesis I've had the pleasure of working with some researchers from whom I learnt a lot. First and foremost I would like to express my sincere gratitude to my supervisor Dr. Elena Fernández for giving me the opportunity to live this great experience, for her unwavering patience, guidance and support.

I would like to extend my thanks to those who also collaborate and contribute in the development of this thesis: thanks to Dr. Jörg Kalscics for his guidance and support over these years and thanks to Dr. Stefan Nickel for his confidence and his support by giving me the opportunity to do part of this work in excellent conditions during my stay in in the Operation Research Department of the Karlsruhe Institute of Technology. I am also deeply grateful to the National Council of Science and Technology in Mexico (CONACYT) for the support throughout these years.

Most of this work was developed in the Departament d'Estadística i Investigació Operativa of the Universitat Politècnica de Catalunya. I want to thank the department and all its members for all the facilities and help I found there, specially to Sonia Navarro and Toni Font for their valuable support. I also want to thank the department PhD students those who shared with lunch times, coffees and more. Special thanks to Daria, Victoria and Jessica, you made the difference in the last months.

I would like to thank Mpaz and Alba for their friendship and fabulous welcome to Barcelona, you both made me feel like home. My deepest thanks to Oriol, for his incessant support, specially for the difficult moments, and for all his encouraging words which kept me going when coffee had lost its stimulating effect. I could not have done it without you. Thanks also to Amadeu, Lluisa and Laia, thanks for being a family for me.

I cannot finish without saying how grateful I am with my very first science mentor, my father, and my best example of strength, my mother. Thanks to Saúl and Itzel as well, for their understanding and for being my deepest strength and motivation. Thanks to all my family for their unconditional support.

Finally, I would like to thank to all people I met during these years, for those who have arrived and for those who have gone, thanks for being part of this adventure.

Abstract

In many situations, the resources in organizations are employed to satisfy some demand (or services) requirements, which are repeated with some periodicity. These recurrent services appear in a large variety of processes such as manufacturing, logistics and several other types of services. This thesis addresses a family of problems that consider services in a recurrent manner. In particular, we concentrate on recurrent service problems with single-period duration with particular emphasis on the study of the strategy that is followed to offer the services over the planning horizon, that is, the scheduling policy.

The aim in this thesis is to study and analyze different options for such policies. The purpose is to provide enough support to decision makers to determine the convenience of using (or not) flexible policies as an alternative to regular strategies. For this, we study alternative models for two different scheduling policies. First, we present several Mixed Integer Linear Programming formulations. We develop two different types of formulations, which can be classified as sparse and dense, according to the type of coefficients matrices they are associated with, respectively. For each type of formulation, we present two versions. In the first one decision variables are associated with individual demand customers whereas in the second one decision variables are associated with classes of customers with similar characteristics. Additionally, for the flexible policy we propose two different formulations suitable for column generation, which are embedded within a branch-and-price framework. All formulations are compared trough extensive computational experience. We also develop a heuristic algorithm suitable for both scheduling policies, which produces good quality solutions for the studied problems, specially for the flexible policy.

Finally, the structure of the solutions obtained with both scheduling policies are analyzed giving important insights on the trade-off between the regular and the flexible policies.

Keywords: combinatorial optimization, mixed-integer programming, multi-period problems, service scheduling, scheduling policies, heuristics, column generation, branch-and-price.

Resumen

En muchas situaciones los recursos en las organizaciones se usan para satisfacer requerimientos de demanda (o servicios) los cuales se repiten con cierta periodicidad. Estos servicios recurrentes suelen aparecer en una gran variedad de procesos de manufactura, logística y varios otros tipos de servicios. Esta tesis aborda una familia de problemas con servicios que se aparecen de manera recurrente. En particular, nos concentramos en problemas de servicio recurrente con duración de un solo periodo haciendo énfasis en el estudio de la estrategia que se sigue para ofrecer los servicios en el horizonte de planeación, es decir, la política de calendarización.

El objetivo de esta tesis es el estudio y análisis de diferentes opciones para este tipo de políticas. El propósito es proporcionar una base suficiente para los tomadores de decisiones para determinar la conveniencia de utilizar (o no) políticas flexibles como alternativa a estrategias regulares. Para ello, se estudian modelos alternativos para dos diferentes políticas de calendarización. En primer lugar, se presentan varias formulaciones de Programación Lineal Entera Mixta en donde desarrollamos dos tipos de formulaciones, las cuales se clasifican en dispersas y densas, segú el tipo de matrices de coeficientes con las cuales están asociadas. Para cada tipo de formulación, se plantean dos versiones. En la primera versión las variables de decisión están asociadas con las demanadas de clientes individuales, mientras que en la segunda las variables de desición se asocian con clases de clientes con características similares. Adicionalmente, para la política flexible se proponen dos diferenes formulaciones adecuadas para generación de columnas, las cuales introducimos dentro de un marco de branch-and-price. Todas las formulaciones son comparadas por medio de una amplia experiencia computacional. También desarrollamos un algoritmo heurístico adecuado para las dos políticas de alendarización, las cuales producen soluciones de buena calidad para los problemas estudiados, especialmente para la política flexible.

Para finalizar, la estructura de las soluciones obtenidas con las dos políticas de calendarización se analizan dando una visión importante de las ventajas entre la política regular y la flexible.

Palabras clave: optimización combinatoria, programación entera mixta, problemas multiperiodo, programación de servicios, políticas de calendarización, heuristicos, generación de columnas, branch-and-price.

Contents

Introduction

In order to ensure competitiveness, companies continuously address the challenge of optimizing their resources. Since the last century, great efforts have been made in order to reduce the utilization of many types of resources such as goods or services, particularly nowadays when the world is more and more concerned about environmental sustainability. It is unanimously agreed that good use of natural, energy and other types of resources plays a core role in the efficiency of processes.

There is a wide range of situations where resources are used to satisfy some demand (or services) requirements, which are repeated with some periodicity. In particular, they appear in various processes such as manufacturing, logistics and several other types of services. Some examples are services for procurement and production, transportation and distribution, as well as information processing and communication. Recurrent services can be found both at strategic and operational planning levels.

Operations Research (OR) is a very versatile discipline that applies analytical methods to identify the best decisions involved in numerous and diverse processes. By employing techniques such as mathematical modelling, statistical analysis, and mathematical optimization, OR provides optimal or near-optimal solutions to complex decision-making processes such as scheduling, facility planning and forecasting and yield management, to mention just a few.

Scheduling determines one of the largest fields of OR, specifically within combinatorial optimization. Scheduling usually deals with tasks that possibly have priority levels, precedence relationships, earliest starting times, and due dates. The objectives for scheduling problems are very diverse. Some of the most frequent ones are the minimization of the completion time of the last task (makespan) or the minimization of the number of tasks completed after their respective due dates.

In this thesis, we focus on a particular family of problems involving the planning of recurrent services. In these problems resources are assigned to offer recurrent services over a planning horizon. Even if these problems can be classified as scheduling problems, this specific characteristic makes them differ from the typical scheduling problems studied in the literature. A very special characteristic of the problems that we study is that services are considered as single-period tasks. That is, the time needed to start and complete a service never exceeds one time period of the planning horizon. Furthermore, we focus on identifying the single periods when each service is repeated within the time horizon, instead of on the sequence according to which the different services are executed along the time horizon.

We concentrate on modelling aspects for recurrent service problems with single-period duration, and on solution techniques for efficiently finding solutions. Particular emphasis is placed on the study of the strategy that is followed to offer the services over the planning horizon, that is, the policy for scheduling. Our aim is to analyze different options for such scheduling policies. The purpose is to provide enough support to decision makers to determine the convenience of using (or not) flexible policies as an alternative to regular strategies. For this, we study alternative models for two different scheduling policies. These models are addressed from a mathematical programming point of view and, therefore, we present several Mixed Integer Linear Programming (MILP) formulations. We develop two different types of formulations: the first type can be seen as a natural initial approach to the problem and produces sparse coefficients matrices whereas the second type is focused on determining the very first service period for each customer and gives dense matrices. For each type of formulation, we present two versions: an extensive and a compact one. In the first one decision variables are associated with individual demand customers whereas in the second one decision variables are associated with classes of customers with similar characteristics. For the regular policy we develop both types of formulations whereas for the flexible policy we only study the extensive formulation. Figure 1 gives a summary on the different formulations developed for both policies. The formulations for each policy are compared trough extensive computational experience.

	Sparse		Dense	
	Extensive	Compact	Extensive	Compact
Regular policy	X	X	X	X
Flexible policy	X	X		

Figure 1: MILP formulations classification for each scheduling policy.

Since the flexible policy results harder to solve than the regular one, we make use of combinatorial optimization techniques that permit alternative solution methods. In particular, we propose two different formulations suitable for column generation (CG). For each formulation we study the pricing subproblem that allows generating new columns, the initialization phase, as well as a procedure to tackle infeasibility issues. Additionally, we apply stabilization procedures in order to avoid the generation of an excessive number of columns. Each CG algorithm is embedded within a branch-and-price (BP) framework, which combines different branching strategies. The BP was implemented for each CG formulation producing very interesting results that we present and analyze.

Heuristics are alternative combinatorial optimization techniques that provide optimal and near optimal values within small computational times. In this thesis we also propose a heuristic algorithm suitable for both scheduling policies. The heuristic produces good quality solutions for the studied problems, specially for the flexible policy.

Finally, we devote a chapter to compare empirically the structure of the solutions obtained with both scheduling policies. This analysis gives an important insight on the trade-off between the regular and the flexible policies.

The contributions of this thesis are summarized below:

- We propose a new family of problems for scheduling recurrent services with single-period tasks.

- For these problems, we study two different scheduling policies, a regular policy that is based on fixed time intervals between two consecutive services and a flexible one that allows for early services. We analyze the solutions for both policies and compare them in order to decide the best strategy to follow.

- We study several alternative MILP formulations for both scheduling policies and we compare their computational performance.

- We develop a branch-and-price algorithm as an alternative method for solving the flexible policy for the problem studied.

- We propose a heuristic algorithm suitable for both policies to provide good quality and quick solutions.

In the following chapters we give the details of the problems we have addressed and the formulations and solution techniques we have developed. The remainder of this document is as follows. In Chapter 1 we give a general overview of scheduling problems in the literature, as well as of the main features of the family of problems addressed in this thesis. We also include a review of potential applications for them. The formal definition of the problems that we study and some basic properties are presented in Chapter 2. In Chapter 3 we study different MILP formulations for the regular scheduling policy. MILP formulations for a flexible scheduling policy are presented in Chapter 4. Following with the flexible policy, we dedicate Chapter 5 to present alternative MILP formulations suitable for column generation. In this chapter we give the details of the proposed column generation algorithm as well as for the branch-and-price framework. Chapter 6 is devoted to the heuristic algorithm proposed for each scheduling policy. We dedicate Chapter 7 to present the results to all the computational experiments we have run with the different formulations and heuristics proposed in this the thesis Finally, Chapter 8 includes the empirical comparison of the structure of the solutions obtained for the studied policies. This comparison provides guidelines for the best scheduling policy to use for the recurrent service problems that we study. Chapter 9 concludes this thesis with some remarks and comments on future research.

Chapter 1

Multi-period Scheduling: State of the Art

In this chapter we introduce the problems that we study and discuss their relation to other problems in the literature. In particular, there is some connection with three type of problems that appear often as a core component in many practical applications from very diverse fields: allocation, assignment and scheduling. The chapter is divided into four sections. The first section gives a general overview of the typical scheduling problems studied in the literature. In the second section, we describe the main elements for the family of problems from which this thesis is derived whereas the third section provides the details and characteristics of the specific problems we study. In the last section we discuss and present a literature review of several applications for these problems.

1.1 Scheduling problems

Scheduling can be seen as the allocation of resources over time to perform a set of tasks. Broadly speaking, managerial decisions address three kinds of questions: (1) What product or service to provide? (2) on what scale should it be provided? and (3) what resources should be made available? Establishing answers to these questions determines the planning phase, whereas the scheduling phase presumes that answers to the above questions already exist and the resource availabilities have already been fixed by the long term commitments of a prior planning decision. Hence, the scheduling phase does not become a concern until some fundamental planning problems are solved (Baker, 1974).

In its turn, the scheduling phase must answer two kinds of questions: what resources should be made allocated? and, when should each task be performed? In other words, the essence of scheduling problems is divided into two different decisions: (1) allocation decisions and (2) sequencing decisions. More precisely, scheduling problems can be decomposed in two parts: a pure scheduling problem, where a start date and an end date have to be decided for each task, and a sequencing problem, where tasks that compete for the use of the same resource only have to be ordered (Lopez and Roubellat, 2013).

Typically, scheduling theory has been associated with mathematical models. Moreover, the development of useful models and techniques has been the continuing interface between theory and practice. The theoretical perspective is mainly a quantitative approach, which begins with a translation of decision-making goals into an explicit objective function and decision-making restrictions into explicit constraints. Ideally, the objective function should account for all costs in the system that depend on scheduling decisions. Unfortunately, such costs are often difficult to measure in practice, or even to identify completely (Baker and Trietsch, 2009). Concerning feasibility constraints, two families of constraints are commonly found in the literature: the limits on the capacity of the available resource, and the technological restrictions on the order in which tasks can be performed.

Indeed, scheduling conforms a wide research field that deals with a large variety of topics and problems. A detailed overview of scheduling models and solution techniques is beyond the scope of this thesis. For up to date comprehensive surveys the reader is addressed to Allahverdi et al. (2008), Atef et al. (2015), Cardoen et al. (2010), Chaudhry and Khan (2015), Deshmane and Pandhare (2015), Gautam et al. (2015), Hartmann and Briskorn (2010), Janiak et al. (2015), Kalra and Singh (2015) or Pinedo (2001, 2005).

1.2 Problems with recurrent services

Recurrent services refer to demand requirements that have to be satisfied with some periodicity. The problems addressed in this thesis consider services in a recurrent or periodic manner. Even if these problems can be classified as scheduling ones, they hold some specific characteristics, which make them differ from the classical scheduling problems in the literature: All services for the recurrent services problems that we study have equal priority levels and no precedence relationships. In the following we give a general overview of the main features of this family of problems.

The allocation of resources for recurrent services may arise in several situations. In some of them more than one entity is responsible for the execution of the services. In a no-collaborative scenario the entities work autonomously, and therefore, they develop individual plans. When no collaboration exists, each entity assigns its own resources to the services it is in charge of. Hence, there are as many single-service problems as entities exist. Instead, in a collaborative scenario the entities involved are committed to work together. Such scenario arises when the entities have some common interests, for example, when entities work together in order to avoid monopolies (Fernández et al., 2009). In a collaborative scenario entities elaborate a joint planning for the assignment of resources, which may imply (or not) the use of shared resources (e.g. Wäldrich et al., 2006). Within a collaborative scenario operative limitations may apply. For instance, entities may not be allowed to offer some specific service or may have a limited number of services that can provide. Other limitations refer to loyalty issues, as it is the case when services can only be performed by a particular entity or group of entities. When some service must be provided by only a particular entity we say that there exists a service-entity loyalty relationship. Figure 1.1 shows scenario classifications for problems with recurrent services.

In this thesis we focus on a special type of services, which are denoted as single-period. The time required to complete one such service does not exceed the duration of the time period. Thus, each service starts and ends during the time period to which it is assigned. This type of services arise in many practical applications and can be of different kinds. Some examples are collection, delivery and replenishment activities, inspections in a job-shop, service and maintenance of technical equipment and transmission of information pages (see Cheung et al., 2005, Karush and Vazsonyi, 1957, Rossi and Braun, 1996). Independently of their collaborative situation, problems with single-period recurrent services may exhibit some other particularities. The resources availability plays a very important role now as there exists the possibility that resources are used to execute more than one activity in a given time period, since we are considering single-period services, In fact, the number of services that a resource can provide in the same time period depends on its capacity and/or on the duration of the services. In particular,

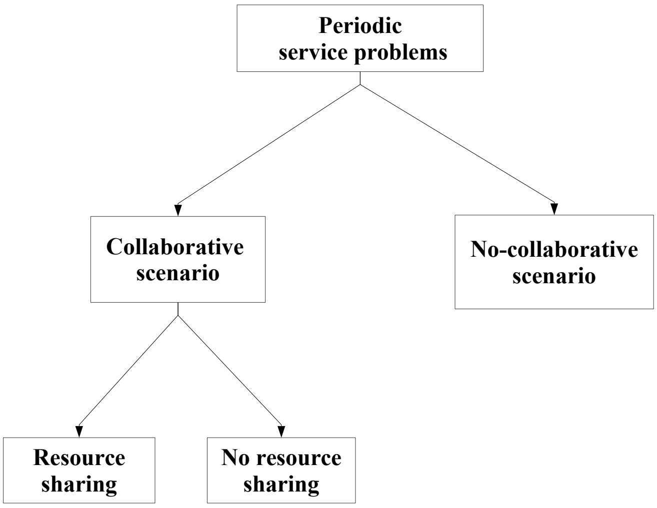


Figure 1.1: Scenarios for recurrent service problems.

the capacities of the resources can be equal (homogeneous) or different (heterogeneous). Furthermore, the duration of the services can be equal (homogeneous) or different (heterogeneous). In the simplest case the time required to complete the services is so short that the total amount of time needed to complete all the services is shorter than the duration of the time period itself. Then, the number of services that a single resource can execute only depends on its capacity. On the contrary, for resources with very large capacities, the number of services provided within a time period will only depend on the duration of the services. A very particular case arises when the capacity of the resources is only limited by the duration of the services. Therefore, an homogeneous service length scenario produces an homogeneous capacity scenario. Conversely, an heterogeneous service length scenario produces an heterogeneous capacity scenario.

In some situations providing services produces some benefits (Liu et al., 2011). In a homogeneous-profit scenario all the services equally contribute to profit. On the contrary, in a heterogeneous-profit scenario different services contribute with different amounts to the overall benefit. Figure 1.2 summarizes the possible alternatives for problems with single-period recurrent services.

Recurrent service problems can be further classified according to their scheduling policy; that is, the strategy followed for scheduling the services. We apply a general classification, which is based on two different schemes: frequency-based and interval-based scheduling policies. Frequency refers, by definition, to the number of occurrences per time unit. Therefore, frequency-based scheduling policies deal with the number of times that a service is provided within the planning horizon (e.g. Bommisetty et al., 1998). This value may be imposed as a fixed number (regular policy) or may be flexible by setting minimum and/or maximum limits (flexible policy). On the other hand, a time interval is, by definition, a metric of the time space between two events of the same nature. Therefore, interval-based scheduling policies deal with the time periods between two consecutive executions of a particular service (Kovalyov et al., 2007). Similarly to the previous scheme, when a regular policy is applied, the time interval between two consecutive services is set to a fixed value whereas in a flexible policy, the time interval between two consecutive executions of a service is restricted within a minimum and/or maximum limit (e.g. Kolen et al., 2007). Figure 1.3 displays the two possible scheduling schemes. Indeed a frequency-based approach can be transformed into an equivalent interval-based policy and vice-versa.

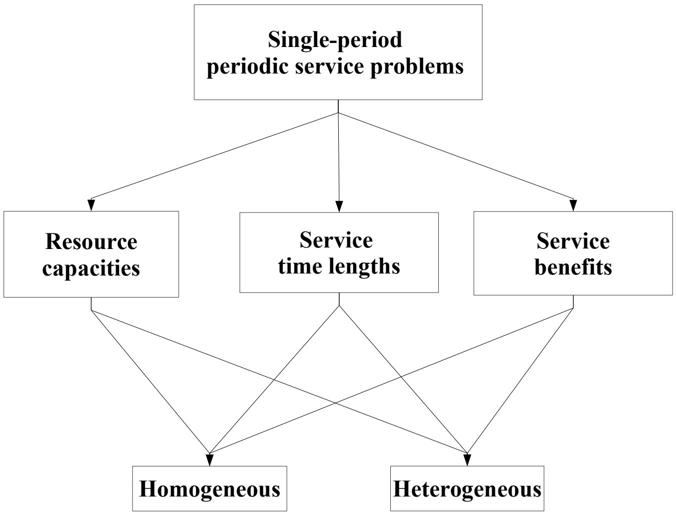


Figure 1.2: Alternatives for single-period recurrent service problems.

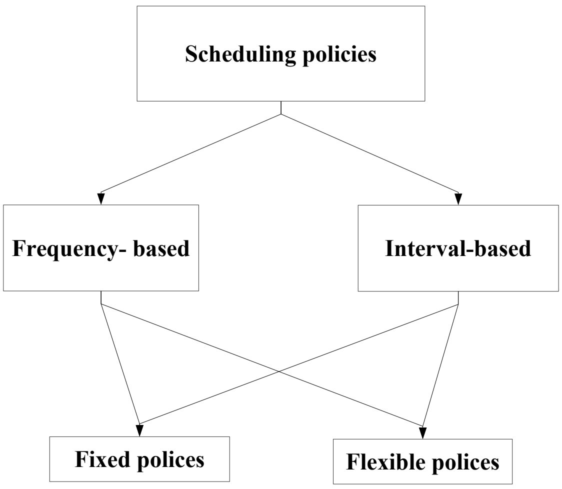


Figure 1.3: Scheduling schemes for recurrent service problems.

In general, finding a way of measuring flexibility in scheduling is a real challenge. In some cases, flexibility is thus considered as a measure of deviation from a given target value (Billaut et al., 2013). This target may be the fixed value considered for regular policies. On the other hand, flexibility may be allowed from different perspectives. One option is to admit positive or negative deviations from the fixed value and to establish thresholds for their possible maximum values. For a frequency-based scheduling policy, the flexibility may be translated as limiting, with lower and/or upper values, the number of times that a service must be provided within the time horizon. For an interval-based scheduling policy, the flexibility may be translated as setting limits for minimum and/or maximum time intervals between consecutive executions of the services.

Another approach to manage flexibility is penalizing the deviations from target values (Lauff and Werner, 2004). For a frequency-based scheduling policy, the flexibility will be translated as a penalty on the surplus or deficit on the number of times that services must be provided within the time horizon. For an interval-based scheduling policy, the flexibility will be translated as a penalty on the deviation from fixed interval values in the actual executions of the services. Under this policy, the measure of flexibility can produce positive or negative values, which are somehow related to the earliness and tardiness concepts of the typical scheduling problems (Choi et al., 1994, Sidney, 1977). Figure 1.2 shows the above described approaches to flexibility for recurrent services.

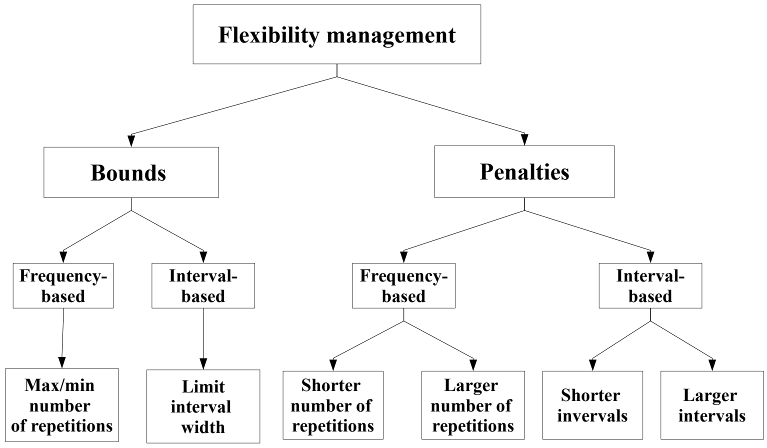


Figure 1.4: Flexibility management for recurrent service problems.

As we will see in the following chapters, the problems that we study are single-period recurrent service problems within a no-collaborative scenario. There is one entity with enough resources, which is the only one responsible for providing the services. All services are of equal length and the capacity of the resources is uniquely determined by the time duration of the services. Hence, resources have homogeneous capacities. For these problems, the execution of services does not provide any type of profit. We focus on two different scheduling policies: one regular and one flexible. Both policies are interval-based, that is, both policies focus on the time interval between two consecutive services. In the regular scheduling policy, we consider that all time intervals between two consecutive executions of a service must be of the same fixed length. In the flexible policy, intervals between consecutive services of lengths larger than some given values are forbidden. Instead, shorter intervals between consecutive services are allowed, although early executions are penalized. Below we give more details of the problems we study.

1.3 Multi-period service scheduling problems

Multi-period service scheduling problems consider single-period recurrent services, and can be described as follows. We are given a finite planning horizon that is partitioned into a fixed set of time periods of equal length. Moreover, we are given a set of customers who periodically have demand of some type of service. These services are translated to visits, which can be any type of collection, delivery, or maintenance activity. Each time a customer has demand in a certain period, we call this a *service request*. We assume that no request spans more than one period and that a customer has not more than one request per period. The time between consecutive service requests can either be fixed or vary over time. Moreover, the duration of each service request within a period is considered of equal size, that is, we consider homogeneous service time lengths. To satisfy the service requests, we are given a set of *servers* or *operators* that can provide the service at the customer location, or remotely. In this problem the capacity of operators depends uniquely on the required time to complete the services. Hereby, each operator has a fixed capacity per time period. Furthermore, a fixed cost is incurred whenever an operator serves to one or more customers in a period. If a customer is serviced by an operator in a given period, we call this a *service period* for the customer and we say that the operator executes a *service visit*. Concerning the occurrence of service requests, instead of predetermined service requests, we assume that there is just a maximum duration between two consecutive service visits, which is specified for each customer. Each time a customer receives service in a period this "generates" a new service request that has to be satisfied within a maximum duration.

In the multi-period service scheduling problems that we study, neither the number of service requests nor the actual periods when they occur are known in advance. Moreover, the satisfaction of a request may incur an additional cost. Concerning the satisfaction of service requests, we consider two different policies. In the first one, each service request has to be satisfied in the period of the request "due date". Hence, given the maximum duration, we schedule the next visit as late as possible. The rationale behind this policy is that it will result in regular visiting schedules for customers, provided that the maximum duration does not change over time. In the second policy, we assume that a customer may be visited *ahead of time*, i.e., before the period or "due date" of the service request. Even if this might increase the total number of visits and result in irregular visiting schedules, it will often allow planners to determine more efficient and better utilized service schedules. However, visits ahead of time may also incur additional costs. Therefore, we have to decide for each customer in which periods we schedule its service visits, in such a way that all customers receive their desired service within the required time and the overall costs for scheduling the operators and satisfying the service requests is minimal.

1.4 Practical applications and related literature

The first and primary application to the multi-period service problems faced in this thesis refers to the logistics of the collection or delivery of commodities, raw materials, or waste. In these problems, customers either produce or consume items at a given rate per period and they can only store a certain amount at their location. Therefore, customers have to be collected or delivered from time to time with an appropriate vehicle and items must be transported to or picked up from some facility, e.g., a warehouse, factory, or recycling center. In this problem the task is to determine in which periods to service each customer such that the storage limitations at the customers are adhered to and as few tours as possible are needed. This problem arises, for example, in the collection and recycling activities of waste of electrical and electronic equipment (WEEE). According to the EU regulation, inhabitants can return their WEEE free of charge at collection points which are run by the local municipalities. Once a storage container- usually an iron-barred box- at a collection point is full, one of the companies selling electrical or electronic products has to organize the pickup of the box and the recycling of its contents. Some examples are presented in Queiruga et al. (2008) and Fernández et al. (2009). As 5-10 iron-barred boxes fit in a truck, the logistics provider can schedule a truck to visit several collection points in a day. As the filling rate of these boxes differs between collection points, one should organize individual routes that allow for irregular visiting schedules for the collection points and aim at maximizing the vehicle utilizations. Herrmann (2011) describes an application where waste has to be periodically collected from waste collection rooms in a healthcare facility.

A similar situation occurs if customers resemble retailers where the stock has to be replenished. In these applications operators correspond to vehicles and tours have to be built such that no stock-outs occur at the retailers and the routing and inventory costs are minimal. Some examples are mentioned in Campbell and Wilson (2014) and Coelho et al. (2013). The operator costs resemble costs for vehicle usage and the service request costs can model inventory costs. In both settings it is possible to schedule a visit to a customer ahead of time to pick up a not yet full container or replenish inventory before reaching the re-order point. Although this will often increase the underlying travel distances, the total number of tours required will be smaller. Some similar decisions are faced in lot-sizing and inventory/routing problems, which deal with variables to control activities between two consecutive periods (see Avella et al., 2015, Coelho and Laporte, 2013, Pochet and Wolsey, 2006, Solyalı and Süral, 2011).

Multi-period service scheduling problems can be seen as a first level decision related to some Vehicle Routing Problems (VRPs), like the periodic VRP (Archetti et al., 2014, Fargeas et al., 2012, Francis and Smilowitz, 2006, Gaudioso and Paletta, 1992, Gonçalves et al., 2005) and the inventory routing problems (IRPs) (Gaur and Fisher, 2004, Le Blanc et al., 2006, Rusdiansyah and Tsao, 2005), when flexible visit frequencies are allowed. It is important to note that in some of these problems, the modeling hypotheses do not coincide with the ones stated in this thesis. One example is the work of Gaudioso and Paletta (1992), which considers a periodic VRP in which the scheduling policy for each customer must be determined, but for each customer the times between two successive visits must be of equal length. Furthermore, the main objective in these models is to minimize the total travel distances or to minimize the fleet size. As can be seen with small examples, minimizing any of these objectives does not necessarily minimize the total number of tours, which is the objective that we consider throughout our work.

Focussing on the tactical modeling aspects of reducing the fleet size and, hence, maximizing the utilization of vehicles coincides with the current trend in vehicle routing problems, not only for economic savings but also due the environmental benefits, see Sbihi and Eglese (2007). Related to this, but without the routing aspect, are joint replenishment problems (cf. Khouja and Goyal, 2008). There are also applications where customers do not receive their service on-the-spot but have to travel to the operators; mainly in the public sector. For example public libraries or banks organized on a truck traveling from community to community, which are still common in rural areas, or medical teams that organize and carry out blood donation sessions. These trucks or teams periodically visit communities. In this way, a reasonable goal is to minimize the total number of in-the-field periods of trucks or teams (cf. Jeffries and O'Hanley, 2012).

Another application is the scheduling of inspectors visits for preventive service and maintenance of technical equipment, e.g. production machines or airplanes. For these regular activities, there is a fixed cost for each service visit and an operational cost that is proportional to the time between two consecutive service visits. The goal is to determine a maintenance schedule minimizing the total costs. Bar-Noy et al. (2002) discuss a machine maintenance problem in which a fixed number of machines can be inspected in each period. The goal is to minimize the total costs, consisting of a fixed cost for carrying out an inspection plus a variable cost that is proportional to the number of periods between two consecutive maintenance periods. The authors showed that the problem is NP-hard and present several approximation algorithms. Moreover, Grigoriev et al. (2006) consider the special case where just one machine can be inspected per period. In this work, different linear and non-linear programming formulations for the problem are presented, as well as an exact solution approach using column generation.

There exist other applications in which, in addition, it has to be ensured that the time between consecutive visits does not exceed a given threshold. Han et al. (1996) discuss the distance-constrained task scheduling problem where the distance-constraints specify the maximal time between two consecutive executions of a task. The authors focus on feasibility aspects and schedulability conditions instead of formulations and solution approaches.

Alternatively, some other authors try to obtain regular schedules by minimizing the variation of the time between consecutive service visits over all customers. Corominas et al. (2010) try to maximize the regularity of the service schedules for each machine by minimizing the variability between consecutive service periods. A linear programming formulation for the problem is discussed and several improvements for it are introduced. In subsequent papers, García-Villoria et al. (2013) present a specially tailored branch-and-bound algorithm and Corominas et al. (2012) propose various heuristics for the same problem. Although the essential task of scheduling service periods is the same, the objectives and/or constraints of the problems addressed in the references above differ from the problems faced in this thesis, especially concerning the maximum duration constraint, rendering the developed solution approaches no longer applicable.

When technicians can maintain more than one machine per time period and they are not committed to working each period on maintenance, a reasonable goal is to minimize, in addition, the total number of in-the-field periods of technicians. A problem with these characteristics is related to sensor scheduling. One example is due to Yavuz and Jeffcoat (2007). There exist some other "non-physical" applications, for example, scheduling problems referring to maintenance and backups of computer systems.

Another example occurs in broadcasting environments where transmissions of information pages have to be periodically scheduled in channels. For each page a maximum time between two consecutive transmissions is given and each channel may broadcast only one page per time unit. The goal is to schedule the transmissions of all pages on a minimum number of channels. This problem is called windows scheduling. Operators coincide with channels and pages with customers. Bar-Noy and Ladner (2003) present an algorithm to construct asymptotically close to optimal schedules for the windows scheduling problem where each page has a unit transmission time. Bar-Noy et al. (2012) introduce a constant approximation scheme for the case of integer length transmission times of pages. They also present a greedy method based on classical bin packing algorithms. Bar-Noy et al. (2007) establish a relation between window scheduling and bin packing. While the window scheduling problem allows the ahead of time scheduling of services, each channel, i.e., operator, may only transmit one page, i.e., customer, per period. Moreover, the focus is on minimizing the maximal number of required channels over all periods. A similar problem arises in media-on-demand systems.

More works related to multi-period service scheduling problems can be found in Campbell and Hardin (2005) and Korst et al. (1994). The former considers a replenishment activity on fixed time intervals. In this work a direct shipping is assumed, i.e., every vehicle can only be assigned to one customer. The objective is to minimize the total fleet size. Korst et al. (1994) address a periodic assignment problem where the executions of periodic operations with fixed start times must be scheduled over an infinite time horizon. The objective is to minimize the number of identical processors needed for the executions.

An alternative view to the problems that we study is that of scheduling problems for multiple visits to customers. In this case processing times are discarded. If we assume that we have a finite set of machines whose capacity is measured as the number of tasks they can process, the objective is to minimize the number of machines simultaneously working in each period. Some related periodic and single period scheduling problems can be found in Chan et al. (2008), Coffman et al. (1978), Korst et al. (1994), Orlin (1982), Park and Yun (1985).

Finally, multi-period service scheduling problems can also be seen as a multi-period multi-machine scheduling problems in which a set of jobs with varying processing times have to be scheduled periodically and there is a finite set of machines with a finite capacity. In addition, it is assumed that, once one of the jobs has been scheduled on a machine in a period, this machine cannot be used in the same period for jobs of the same set (e.g., because of too large set-up times). Then, a reasonable objective is to minimize the total number of machines used in all periods, i.e., the total number of blocked “machine-periods”.

To the best of our knowledge, the multi-period service scheduling problems that we study have not been yet been addressed in the literature.

Chapter 2

Problem Definition

In this thesis we study a multi-period service problem, which was originally motivated by the WEEE application presented in Chapter 1.4. In this problem we consider interval-based scheduling policies, and we assume that the maximum duration between two consecutive service visits for each customer is known in advance and independent from time. Moreover, there is no fundamental distinction between customers service requests so any of them can be satisfied by any operator. In addition, a customer can be visited by a different operator each time. We want to determine a service schedule such that all service requests are satisfied and the total costs for using operators and ahead of time visits are minimal. We name this problem as the Multi-period Service Scheduling Problem (MSSP).

A first aim of this work is to compare two different policies for scheduling service visits for the MSSP. In particular, there is a special interest in the trade-off between efficiency, measured as the total number of operators needed, and regularity, measured as a function of the total number of periods when visits are scheduled ahead of time. A second objective is to propose alternative ways of solving the MSSP, for both considered scheduling policies.

In this chapter we give the formal definition for the MSSP. Moreover, we give the details for each of the scheduling policies we consider by describing the two versions for the problem. We include a comprehensive example to better perceive the practical implications between the scheduling. The chapter ends with some basic properties for the MSSP.

2.1 Formal Definition

We denote by T the index set of (discretized) time periods (or simply, periods). Thus, $|T|$ denotes the length of the time horizon, measured in number of periods. We assume that all time periods have the same length. The index set of customers is denoted by I and the maximum duration between two consecutive visits to customer $i \in I$, measured in periods, by $t_i \in \mathbb{N}$. t_i is also called the *service interval* of i. The index set of operators is denoted by K. The maximal number Q of customers an operator can serve per period is fixed and identical for all operators. We assume that all customers have been served just before the start of the planning horizon.

For a customer i, a *calendar* $C_i \subseteq T$ is an ordered sequence of periods such that the number of periods between any two consecutive elements in C_i does not exceed the service interval t_i. The periods of a given calendar of a customer are referred to as *service periods* of the customer. A *service cluster* consists of a set of customers visited by the same operator in the same period. The size of service clusters must not exceed the capacity of the operator. We say that a customer is *scheduled* (*unscheduled*) if we have already (not yet) determined a calendar for him. Because different customers may have different service intervals, the number of operators to be used along the time horizon is established on a per period basis. This means that we do not assume that regular schedules should apply. Therefore, a specific service cluster may be formed just once throughout the time horizon and service clusters may be different for each period of the time horizon.

In the MSSP the following decisions must be made:

- Determine a calendar for each customer, i.e., the set of periods when each customer will be visited. That is, determine a schedule $S = \{C_i : i \in I\}$.
- Assign each service period of a customer to an operator. This assignment must take into account the capacity of the operators.

2.2 Scheduling policies

In the MSSP we focus on modelling aspects aimed at reducing the total number of operators. Thus, we consider two alternative policies for the schedules. Both policies are described below.

2.2.1 Periodic service policy

In the periodic service (PS) policy, the time intervals between two consecutive service periods of a customer are always of the same length and coincide with its service interval. With this policy, once the first service period is decided for a given customer, subsequent service periods for this customer are uniquely determined. Since we are not assuming that all customers have the same service intervals, even if each customer is visited periodically, this policy does not produce identical service clusters. The goal with this policy is thus to minimize the total number of operators used over all periods of the planning horizon (z). We call the MSSP with a periodic service policy the Periodic MSSP, for short P-MSSP.

2.2.2 Aperiodic service policy

In the aperiodic service (AS) policy, customers may be visited ahead of time. This means that the time intervals between two consecutive service periods of a customer are not established in advance and can be of different lengths. With this policy, it is still not allowed that the time between two consecutive visits to the same customer exceeds his or her service interval. However, it is permitted to visit a customer before the end of the service interval. If the number of periods h between two consecutive service periods for customer i is smaller than his service interval t_i, then we call this an early visit and $t_i - h$ the *earliness* of the visit. The earliness of a schedule is the total earliness of all visits in the calendars and the earliness of a customer is the earliness of his calendar. The goal with this policy is then to minimize a weighted sum of the total number of operators used over all periods of the time horizon (z) and the total earliness of all customers (e). Both criteria are considered within different scenarios by assigning different values to a weight parameter $\beta \in [0, 1]$. In particular, the solution value for a schedule with this policy is defined as:

$$f(S) = \beta z + (1 - \beta)e \tag{2.1}$$

The MSSP with aperiodic service policy is referred to as Aperiodic MSSP, for short A-MSSP.

In the following, periodic and aperiodic service policies will be simply denoted as PS and AS policies, respectively. The rationale behind the AS policy is that allowing early visits may reduce the overall number of operators needed throughout the time horizon, by making a better use of the operator capacities. Of course, allowing early services may cause a negative effect since customers are not collected periodically (thus, possibly needing additional planning at the customers).

2.3 Example

We next present a small example to illustrate the MSSP and to highlight the utility of the AS policy. Suppose there are four customers $I = \{1, 2, 3, 4\}$ to be visited in a time horizon of $|T| = 12$ periods. The service intervals of the customers are $t_1 = 2$, $t_2 = 3$, $t_3 = 4$ and $t_4 = 3$ periods, respectively. There are $|K| = 2$ operators available, where each one can serve $Q = 3$ customers per period. Figure 2.1 displays schedule S_A, which is a feasible solution for the P-MSSP and A-MSSP, and optimal for the P-MSSP.

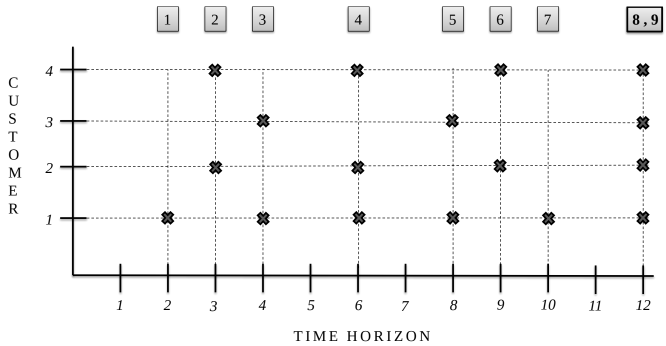


Figure 2.1: Schedule solution S_A.

For schedule S_A, the calendars for the customers are: $C_1 = \{2, 4, 6, 8, 10, 12\}$, $C_2 = \{3, 6, 9, 12\}$, $C_3 = \{4, 8, 12\}$, $C_4 = \{3, 6, 9, 12\}$. Since $Q = 3$, we need one operator for each of the periods *2, 3, 4, 6, 8, 9*, and *10*, and two operators for period *12*. Hence, we need in total $z_A = 9$ operators over the twelve periods.

If, however, the AS policy is applied, schedule S_B of Figure 2.2 is also a feasible solution for the A-MSSP. For schedule S_B, the calendars for the customers are: $C_1 = \{2, 4, 6, 8, 9, 11\}$, $C_2 = \{3, 6, 9, 12\}$, $C_3 = \{4, 8, 12\}$, $C_4 = \{3, 6, 9, 12\}$.

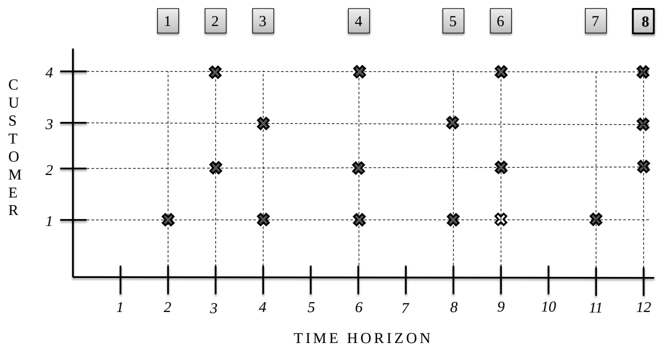


Figure 2.2: Schedule solution S_B.

We can observe that for schedule S_B we just need $z_B = 8$ operators instead of nine. That is, we save an entire operator by moving the service period of customer 1 from period *10* to period *9*. Hence, we visit customer 1 one period before its next designated service period. Therefore, the operator scheduled to serve just one customer in period *10* is saved with a total earliness of $e_B = 1$.

The optimality for schedule S_B for the A-MSSP depends, however, on the parameter β. Table 2.1 displays the solution value for schedules S_A and S_B for different values of β. We can observe that, for $\beta = 0.1$, $f(S_A) = 0.9 < 1.7 = f(S_B)$, whereas for $\beta = 0.9$, $f(S_A) = 8.1 > 7.3 = f(S_B)$. In fact, S_A is an optimal solution for the A-MSSP with $\beta = 0.1$. On the contrary, S_B is an optimal solution for the A-MSSP with $\beta = 0.9$.

β	S_A			S_B		
	βz_A	$(1-\beta)e_A$	$f(S_A)$	βz_B	$(1-\beta)e_B$	$f(S_B)$
0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.1	0.9	0.0	0.9	0.8	0.9	1.7
0.2	1.8	0.0	1.8	1.6	0.8	2.4
0.3	2.7	0.0	2.7	2.4	0.7	3.1
0.4	3.6	0.0	3.6	3.2	0.6	3.8
0.5	4.5	0.0	4.5	4.0	0.5	4.5
0.6	5.4	0.0	5.4	4.8	0.4	5.2
0.7	6.3	0.0	6.3	5.6	0.3	5.9
0.8	7.2	0.0	7.2	6.4	0.2	6.6
0.9	8.1	0.0	8.1	7.2	0.1	7.3
1.0	9.0	0.0	9.0	8.0	0.0	8.0

Table 2.1: Solution values for the A-MSSP for different values of β.

2.4 Properties

Below we discuss some properties of the P-MSSP and the A-MSSP. First we introduce some additional notation. For customer $i \in I$, we define by $T_i^1 \in \{1, \ldots, t_i\}$ as the first service period of customer i. Thus, the calendar of customer i can be denoted as $C_i = \{T_i^1, \ldots, T_i^{n_i^*}\}$, where n_i^* is the maximum number of service periods of customer i. We assume that these service periods are ordered by increasing values, i.e., ${T_i}^1 < {T_i}^2 < \ldots < {T_i}^{n_i^*}$.

Property 1. *A calendar is feasible for customer $i \in I$ for the PS policy, if $T_i^n = T_i^{n-1} + t_i$, for $n \in \{2, \ldots, n_i^*\}$, with $n_i^* = 1 + \left\lfloor \frac{|T| - T_i^1}{t_i} \right\rfloor$. For the AS policy, a calendar is feasible for customer i if $T_i^n = T_i^{n-1} + h_i^n$, with $h_i^n \in \{1, \ldots, t_i\}$, $n \geq 2$ with $T^n \leq |T|$.*

Property 2. *If $Q < |I|$, and $|K|$ is small, it is possible to build instances for the P-MSSP and A-MSSP that are infeasible. A sufficient condition that guarantees that an instance is feasible for both P-MSSP and A-MSSP is $|K| \geq \left\lceil \frac{|I|}{Q} \right\rceil$.*

Property 3. *An upper bound on the total number of operators used over all periods of the planning horizon for the PS policy is obtained when all customers are served for the first time at time period 1, i.e. $T_i^1 = 1$, $i \in I$. In this case, the set of customers served at time period t is $I^t = \{i \in I \mid t = 1 + kt_i \text{ for some } k \geq 0\}$. Then, the upper bound is given by:*

$$z = \sum_{t \in T} \left\lceil \frac{|I^t|}{Q} \right\rceil .$$

Since feasible solutions to the P-MSSP are also feasible to the A-MSSP and have 0 earliness, the above solution also gives a valid upper bound to the A-MSSP of value βz. Similar bounds can be obtained by fixing the first visit to all customers at any time period $t \leq \hat{t} = \min_{i \in I} t_i$.

Property 4. *If $\beta = 0$, any feasible solution to the P-MSSP is optimal for the A-MSSP, as its objective function value for the A-MSSP is zero. Hence, the set of optimal solutions to the A-MSSP coincides with the set of feasible solution to the P-MSSP.*

Property 5. *If $|I| = 1$, each visit "consumes" one server, independently of the value of Q. Thus, earliness does not reduce the number of servers needed so the set of optimal solutions for the P-MSSP and the A-MSSP coincide, independently of the value of β. In particular, the solution that visits the customer for the first time in period $T_1^1 = t_1$ and then in periods $T_1^1 + kt_1$, with $k = 1, \dots, \left\lceil \frac{T-t_1}{t_1} \right\rceil$ is optimal. Its objective function value for both P-MSSP and the A-MSSP is $\left\lceil \frac{T-t_1}{t_1} \right\rceil$.*

Property 6. *If $t_i = \bar{t}$, for all $i \in I$, then the set of optimal solutions for the P-MSSP and the A-MSSP coincide, independently of the value of β. Again in this case, earliness cannot reduce the number of servers used when all customers are visited with no earliness in time periods $T_i^1 = \bar{t}$, $T_i^1 + k\bar{t}$, with $k = 1, \dots, \left\lceil \frac{T-t_i}{t_i} \right\rceil$. Such a solution is optimal and its objective function value is $\left\lceil \frac{|I|}{Q} \right\rceil \left\lceil \frac{T-\bar{t}}{\bar{t}} \right\rceil$.*

Property 7. *If $Q = 1$ each visit "consumes" one server, independently of the value of $|I|$. Again, in this case earliness will not reduce the number of servers and the set of optimal solutions for the P-MSSP and the A-MSSP coincide, independently of the value of β. Now the solution where each customer $i \in I$ is visited for the first time in period $T_i^1 = t_i$ and then in periods $T_i^1 + kt_i$, with $k = 1, \dots, \left\lceil \frac{T-t_i}{t_i} \right\rceil$ is optimal. Its value is $\sum_{i \in I} \left\lceil \frac{T-t_i}{t_i} \right\rceil$.*

Property 8. *If $Q \geq |I|$ and for all $i \in I$, t_i is multiple of $\hat{t} = \min_{i \in I} t_i$, then set of optimal solutions to the P-MSSP and the A-MSSP coincide, independently of the value of β. When $Q \geq |I|$, any schedule where the calendars respect the service intervals of the customers is feasible. Moreover, any schedule that minimizes the number of time periods when services occur will be optimal to the P-MSSP (as it would use just one server at any "busy" time period). Since, for all $i \in I$ t_i is multiple of $\hat{t}$, such an optimal schedule will serve customers only at time periods t multiple of $\hat{t}$. The above schedule is also optimal to the A-MSSP, as it minimizes the overall number of servers and its earliness is zero.*

Chapter 3

Formulations for the P-MSSP

In this chapter, we present four alternative MILP formulations for the P-MSSP. We have classified these formulations into two different types: sparse and dense, according to the type of coefficients matrices to which they are associated. For each type, we develop two variants, which we refer to as *customer-based* and *class-based* formulations. In the following sections we give the details for these formulations. In Chapter 7.2, we present and analyze the numerical results of the computational experiments we have run. We also include a comparison of the computational performance of the formulations.

3.1 Sparse formulations

The sparse MILPs we present below are a first approach to a formal statement of the P-MSSP. That is, we base the formulations on decision variables that determine the allocation of customers to operators at all time periods. For the MSSP, these MILP formulations are associated with sparse coefficient matrices.

3.1.1 A customer-based sparse

The very first formulation we propose for the P-MSSP uses the following sets of binary decision variables:

For $i \in I$, $t \in T$,

$$x_i^t = \begin{cases} 1 & \text{if customer } i \text{ is visited in period } t \\ 0 & \text{otherwise} \end{cases}$$

For $i \in I$, $k \in K$, $t \in T$,

$$y_{ik}^t = \begin{cases} 1 & \text{if customer } i \text{ is visited by operator } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

For $k \in K$, $t \in T$,

$$z_k^t = \begin{cases} 1 & \text{if operator } k \text{ is used in period } t \\ 0 & \text{otherwise} \end{cases}$$

The customer-based sparse formulation for the P-MSSP is as follows:

$$(PS) \qquad \min \sum_{t\in T}\sum_{k\in K} z_k^t \tag{3.1}$$

$$s.t. \quad \sum_{t=1}^{t_i} x_i^t = 1 \qquad i \in I \tag{3.2}$$

$$x_i^t = x_i^{t+t_i} \qquad i \in I, t \in \{1, ..., |T| - t_i\} \tag{3.3}$$

$$x_i^t = \sum_{k\in K} y_{ik}^t \qquad i \in I, t \in T \tag{3.4}$$

$$\sum_{i\in I} y_{ik}^t \le Q z_k^t \qquad k \in K, t \in T \tag{3.5}$$

$$Q z_k^t \le \sum_{i\in I} y_{i,k-1}^t \qquad k \in K \setminus \{1\}, t \in T \tag{3.6}$$

$$x_i^t,\ z_k^t,\ y_{ik}^t \in \{0, 1\}, \qquad i \in I, k \in K, t \in T \tag{3.7}$$

The objective (3.1) minimizes the total number of operators needed for the service throughout the time horizon. The first visit for each customer is established by constraints (3.2), whereas consecutive service periods throughout the time horizon are imposed by constraints (3.3). Constraints (3.4) guarantee that if a customer is visited at period t, then it is assigned to some operator in that period. Constraints (3.5) are capacity constraints that ensure that the number of customers assigned to each operator must not exceed the operator capacity. Constraints (3.6) are symmetry breaking constraints which imposes that at each period t, operator k will not be required unless operator $1, \ldots, k-1$ are full, i.e., each of them has Q assigned customers. Note that these constraints are valid because the objective is to minimize the total number of operator. Finally, Constraints (3.7) enforce the variables to be binary.

3.1.2 A class-based sparse

The formulation below is based on the observation that the service intervals of several customers may coincide. If we classify all customers according to their service intervals, we obtain groups (or classes) of customers instead of single customers. This formulation takes advantage of such classes of customers. In the following, the formulations that make use of classes instead of single customers are referred to as class-based formulations. Below we describe the details for such formulations, and we present its version for the sparse formulation for the P-MSSP.

According to the different values of the service intervals, t_i, we classify customers $i \in I$ into *interval classes* (or simply, *classes*). That is, customers with equal service interval belong to the same class. If the service intervals were the same for all customers, we would have a single class of size $|I|$. On the contrary, if the service intervals were all different, we would have $|I|$ classes, each of size one. We define J as the set of indices for the classes, where $|J| \le |I|$. For each class $j \in J$, u_j denotes the service interval for class j, i.e., the common service interval for all customers of class j, and w_j, the size of class j, i.e., the number customers with common service interval equal to u_j.

The class-based sparse formulation for the P-MSSP uses the following sets of decision variables:

For $j \in J$, $t \in T$,

x_j^t = number of customers of class j that are visited in period t.

For $j \in J$, $k \in K$, $t \in T$,

y_{jk}^t = number of customers of class j that are visited by operator k in period t.

For $k \in K$, $t \in T$,

$$z_k^t = \begin{cases} 1 & \text{if operator } k \text{ is used in period } t \\ 0 & \text{otherwise} \end{cases}$$

The class-based sparse formulation for the P-MSSP is as follows:

$$(PS^c) \qquad \min \sum_{t \in T} \sum_{k \in K} z_k^t \tag{3.8}$$

$$\begin{aligned}
s.t. \quad & \sum_{t=1}^{u_j} x_j^t = w_j && j \in J && (3.9)\\
& x_j^t = x_j^{t+u_j} && j \in J, t \in \{1, ..., |T| - u_j\} && (3.10)\\
& x_j^t = \sum_{k \in K} y_{jk}^t && j \in J, t \in T && (3.11)\\
& \sum_{j \in J} y_{jk}^t \leq Q z_k^t && k \in K, t \in T && (3.12)\\
& Q z_k^t \leq \sum_{j \in J} y_{j,k-1}^t && k \in K \setminus \{1\}, t \in T && (3.13)\\
& z_k^t \in \{0, 1\}, && k \in K, t \in T && (3.14)\\
& x_j^t,\ y_{jk}^t \quad \text{integer}, && j \in J, k \in K, t \in T && (3.15)
\end{aligned}$$

The objective of PS^c minimizes the total number of operators needed for the service throughout the time horizon. Constraints (3.9) guarantee that all the customers of each class are visited for the first time within their service interval, whereas consecutive visit periods throughout the time horizon are determined by Constraints (3.10). Constraints (3.11) guarantee that customers of class j visited at period t are assigned to some operator in that period. Constraints (3.12) are capacity constraints that ensure that the number of customers assigned to each operator does not exceed the operator capacity. Constraints (3.13) are symmetry breaking constraints already explained. Again, these constraints are valid because the objective is to minimize the total number of operators. Finally, Constraints (3.14) enforce variables z_k^t to be binary, while constraints (3.15) enforce the integrality of variables x_j^t and y_{jk}^t.

PS^c has same structure as PS of Section 3.1.1. The difference between both formulations relies on the use of customer classes J instead of customers I, changing accordingly the x and y variables from binary to general integer and reducing therefore the number of decision variables, at the expenses of changing their domain to that of general integer values. When $J = I$, PS and PS^c are the same.

3.2 Dense formulations

The dense formulations for the P-MSSP are based on the core idea that the leading decision for the periodic policy is to determine the period in which customers are visited for the first time over the time horizon. For customer i, the very first service visit must occur within the initial t_i periods. Given that we assume a periodic policy, if customer i is first visited in period $r_i \in R_i$, with $R_i = \{1, \ldots, t_i\}$, his consecutive service periods will occur on periods $r_i + t_i$, $r_i + 2t_i$,..., $r_i + nt_i$, with $n = \left\lceil \frac{|T|-r_i}{t_i} \right\rceil$. Therefore, the service periods for customer i will be all periods t such that $t - r_i$ are multiple of t_i.

3.2.1 A customer-based dense

Below we present a customer-based dense formulation. This formulation uses the following sets of decision variables:

For $i \in I$, $r \in \{1, \ldots, t_i\}$,

$$x_i^r = \begin{cases} 1 & \text{if customer } i \text{ is visited for the first time in period } r \\ 0 & \text{otherwise} \end{cases}$$

For $t \in T$,

$z^t =$ the number of operators needed in period t.

The customer-based dense formulation for the P-MSSP is as follows:

$$(PD) \qquad \min \sum_{t \in T} z^t \tag{3.16}$$

$$\begin{aligned}
s.t. \quad & \sum_{r \le t_i} x_i^r = 1 & & i \in I & (3.17)\\
& \sum_{i \in I} \sum_{\substack{r \le t_i \\ t-r=t_i}} a_i^{rt} x_i^r \le Q z^t & & t \in T & (3.18)\\
& x_i^r \in \{0,1\}, & & i \in I, r \in R & (3.19)\\
& z^t \quad \text{integer}, & & t \in T & (3.20)
\end{aligned}$$

In formulation PD, the objective (3.16) minimizes the total number of operators needed for providing service throughout the time horizon. The first visit for each customer is established by constraints (3.17), whereas the consecutive service periods throughout the time horizon are implicitly imposed by constraints (3.18). In particular, the coefficient $a_i^{rt} = 1$, if $t - r$ is multiple of t_i, and $a_i^{rt} = 0$, otherwise. Constraints (3.18) operate also as capacity constraints since they ensure that the number of customers assigned to operators in period t does not exceed the operators capacity. Finally, constraints (3.19) enforce the variables x_i^r to be binary and constraints (3.20) enforce the integrality of variables z^t.

3.2.2 A class-based dense

An alternative dense formulation for the P-MSSP can be derived by applying the *class-based* idea described in Section 3.1.2. In the class-based version of the dense formulation, the first visit to customers in class j must occur within the initial u_j periods. Given that we are assuming periodic policy, if any customer of class j is first visited in period $r_j \in R_j$, with $R_j = \{1, \dots, u_j\}$, his consecutive service periods will occur in periods $r_j + u_j$, $r_j + 2u_j$,..., $r_j + nu_j$, with $n = \left\lceil \frac{|T| - r_j}{u_j} \right\rceil$. Therefore, the service periods for a customer of class j will be all periods t such that $t - r_j$ are multiple of u_j. In particular, the dense class-based formulation use the following sets of decision variables:

For $j \in J$, $r \in \{1, \dots, u_j\}$,

x_j^t = the number of customers of class j that are visited for the first time in period r.

For $t \in T$,

z^t = the number of operators needed in period t.

The class-based dense formulation for the P-MSSP is as follows:

$$PD^c \qquad \min \sum_{t \in T} z^t \tag{3.21}$$

$$s.t. \quad \sum_{r \leq u_j} x_j^r = w_j \qquad j \in J \tag{3.22}$$

$$\sum_{i \in I} \sum_{\substack{r \leq u_j \\ t - r = u_j}} a_j^{rt} x_j^r \leq Q z^t \qquad t \in T \tag{3.23}$$

$$x_j^r,\ z^t \quad \text{integer}, \qquad j \in J, r \in R, t \in T \tag{3.24}$$

In formulation PD^c, objective (3.21) minimizes the total number of operators needed for the services throughout the time horizon. Constraints (3.22) are the same as (3.9) and impose that all the customers of each class are visited for the first time within their service interval. Similarly to formulation PD, the consecutive service periods throughout the time horizon are implicitly imposed by Constraints (3.23), in which now the coefficient $a_j^{rt} = 1$, if $t - r$ is multiple of u_j, and $a_j^{rt} = 0$, otherwise. Again, Constraints (3.23) work also as capacity constraints. Finally, Constraints (3.24) enforce the integrality of the variables.

PD^c has same structure as PD of Section 3.2.1. The difference between formulations relies on the use of the set of classes of customers J instead of the set of customers I, changing accordingly the x variables from binary to general integer variables and reducing therefore the number of such variables. When $J = I$, PD and PD^c are the same.

3.3 Size of the formulations

We devote this section to compare of the dimensions of the P-MSSP formulations presented in Sections 3.1 and 3.2. Table 3.1 displays the total number of variables (n), constraints (m) and non-zero elements in the coefficients matrix (*nonzero*) for each of the P-MSSP formulations. The density for each coefficient matrix is computed as $d = \frac{nonzero}{n}$. We can observe the density difference between sparse (PS, PS^c) and dense (PD, PD^c) formulations.

Formulation	n	m	$nonzero$	d
PS	$\lvert T\rvert\left[\lvert I\rvert\left(\lvert K\rvert+1\right)+\lvert K\rvert\right]$	$\lvert I\rvert-\sum_{i\in I}t_i+\lvert T\rvert\left(2\lvert I\rvert+2\lvert K\rvert-1\right)$	$\lvert I\rvert\lvert T\rvert\left(3\lvert K\rvert+2\right)+2\lvert T\rvert\lvert K\rvert-\lvert T\rvert-\sum_{i\in I}t_i$	$\frac{\lvert I\rvert\lvert T\rvert(3\lvert K\rvert+2)+2\lvert T\rvert\lvert K\rvert-\lvert T\rvert-\sum_{i\in I}t_i}{\lvert T\rvert[\lvert I\rvert(\lvert K\rvert+1)+\lvert K\rvert]}$
PS^c	$\lvert T\rvert\left[\lvert J\rvert\left(\lvert K\rvert+1\right)+\lvert K\rvert\right]$	$\lvert J\rvert-\sum_{j\in J}u_j+\lvert T\rvert\left(2\lvert J\rvert+2\lvert K\rvert-1\right)$	$\lvert J\rvert\lvert T\rvert\left(3\lvert K\rvert+2\right)+2\lvert T\rvert\lvert K\rvert-\lvert T\rvert-\sum_{j\in J}u_j$	$\frac{\lvert J\rvert\lvert T\rvert(3\lvert K\rvert+2)+2\lvert T\rvert\lvert K\rvert-\lvert T\rvert-\sum_{j\in J}u_j}{\lvert T\rvert[\lvert J\rvert(\lvert K\rvert+1)+\lvert K\rvert]}$
PD	$\lvert T\rvert+\sum_{i\in I}t_i$	$\lvert I\rvert+\lvert T\rvert$	$\lvert T\rvert-\sum_{i\in I}t_i+\lvert T\rvert\sum_{i\in I}t_i+\lvert T\rvert\sum_{i\in I}\sum_{r=1}^{t_i}\left\lfloor\frac{\lvert T\rvert-r}{t_i}\right\rfloor$	$\frac{\lvert T\rvert-\sum_{i\in I}t_i+\lvert T\rvert\sum_{i\in I}t_i+\lvert T\rvert\sum_{i\in I}\sum_{r=1}^{t_i}\left\lfloor\frac{\lvert T\rvert-r}{t_i}\right\rfloor}{\lvert T\rvert+\sum_{i\in I}t_i}$
PD^c	$\lvert T\rvert+\sum_{j\in J}u_j$	$\lvert J\rvert+\lvert T\rvert$	$\lvert T\rvert-\sum_{j\in J}u_j+\lvert T\rvert\sum_{j\in J}u_j+\lvert T\rvert\sum_{j\in J}\sum_{r=1}^{u_j}\left\lfloor\frac{\lvert T\rvert-r}{u_j}\right\rfloor$	$\frac{\lvert T\rvert-\sum_{j\in J}u_j+\lvert T\rvert\sum_{j\in J}u_j+\lvert T\rvert\sum_{j\in J}\sum_{r=1}^{u_j}\left\lfloor\frac{\lvert T\rvert-r}{u_j}\right\rfloor}{\lvert T\rvert+\sum_{j\in J}u_j}$

Table 3.1: Matrices element comparison for the P-MSSP formulations.

Chapter 4

Formulations for the A-MSSP

In the AS policy, the first visit of a customer is not enough to determine the subsequent visits during the time horizon. The only information we have is that if customer i is visited at period t, then it must be visited again no more than $m_i^t = \min\{t_i, |T| - t\}$ periods later, but we do not know the specific period when it will be visited again within the time interval $[t+1, t+m_i^t]$. In order to reduce the negative effect that early services may cause, for the A-MSSP, we minimize not only the total number of operators but also the total *earliness*. We define earliness as the number of early periods within the visits of the customers.

Below we present two alternative MILPs for the A-MSSP which are classified as sparse. For this, we introduce variables that identify the number of periods between two consecutive visits at a given customer. We have developed two variants, which we refer to as customer-based and class-based. We study their properties and compare them computationally. The numeral results are presented in Chapter 7.3.

4.1 Sparse formulations

The sparse MILPs represent a first approach to a formal statement of the A-MSSP. Similarly to the sparse P-MSSP formulations of Chapter 3.1, the sparse versions of the A-MSSP we introduce below use decision variables that determine the customers that are visited in each period, and the operator that provides service to each customer served at a given period. For the MSSP, these formulations are also associated with sparse coefficient matrices.

4.1.1 A customer-based sparse formulation

In addition of the variables defined in Section 3.1.1, we define variables to identify the number of periods between two consecutive visits at a given customer. Let H_i^t be the set of potential periods to schedule a visit for customer $i \in I$ after period $t \in \{0, ..., |T| - 1\}$, i.e., $H_i^t = \{1, \ldots, m_i^t\}$. Therefore, for $i \in I$, $t \in \{0, \ldots, |T| - 1\}$, $h \in H_i^t$ we define

$$f_i^{th} = \begin{cases} 1 & \text{if } t \text{ and } t+h \text{ are consecutive service periods for customer } i, \\ 0 & \text{otherwise} \end{cases}$$

The customer-based MILP for the A-MSSP uses the sets of binary decision variables defined in Section 3.1.1 plus the set of binary variables f_i^{th} explained above. The formulation is the following:

$$(AS_0) \quad \min \ \beta \sum_{t\in T}\sum_{k\in K} z_k^t + (1-\beta)\sum_{i\in I}\sum_{t=1}^{|T|-1}\sum_{h=1}^{m_i^t}(t_i - h) f_i^{th} \tag{4.1}$$

$$s.t. \quad \sum_{h=1}^{t_i} f_i^{0h} \geq 1 \qquad i \in I \tag{4.2}$$

$$x_i^t \leq \sum_{h=1}^{t_i} x_i^{t+h} \qquad i \in I, t \in \{0,...,|T|-t_i\} \tag{4.3}$$

$$x_i^t = \sum_{h=1}^{t_i} f_i^{th} \qquad i \in I, t \in \{0,...,|T|-t_i\} \tag{4.4}$$

$$\sum_{h=1}^{m_i^t} f_i^{th} \leq x_i^t \qquad i \in I, t \in \{|T|-t_i+1,...,|T|-1\} \tag{4.5}$$

$$x_i^t + x_i^{t+h} \leq \sum_{s=1}^{h-1} x_i^{t+s} + f_i^{th} + 1 \qquad i \in I, t \in \{0,...,|T|-1\}, h \in H_i^t \tag{4.6}$$

$$z_k^t \leq \sum_{i\in I} y_{ik}^t \qquad k \in K, t \in T \tag{4.7}$$

$$x_i^t = \sum_{k\in K} y_{ik}^t \qquad i \in I, t \in T \tag{4.8}$$

$$\sum_{i\in I} y_{ik}^t \leq Q z_k^t \qquad k \in K, t \in T \tag{4.9}$$

$$Q z_k^t \leq \sum_{i\in I} y_{i,k-1}^t \qquad k \in K \setminus \{1\}, t \in T \tag{4.10}$$

$$x_i^t, z_k^t, y_{ik}^t, f_i^{th} \in \{0,1\} \qquad i \in I, k \in K, t \in T, h \in H_i^t \tag{4.11}$$

In AS_0 the objective (4.1) minimizes a weighted sum of the total number of operators used in the time horizon and the total earliness. By assigning different values to $\beta \in [0,1]$, both criteria can be considered within different scenarios. Constraints (4.2) guarantee that the first visit period for each customer occurs within his service interval. Constraints (4.3) are scheduling constraints, which ensure that the number of periods between two consecutive visits to the same customer never exceeds his service interval. Constraints (4.4) and (4.5) are logical constraints, which relate the x and f variables. In particular, Constraints (4.4) force that if customer i is visited at period t, then it is revisited for the first time at most t_i periods after period t. Constraints (4.6) are logical constraints, which ensure the variable f_i^{th} to be one only if the first visit of customer i after period t is period $t+h$, i.e., if customer i has been visited at periods t and $t+h$ and there is no other visit for i between periods t and $t+h$. Because of the objective function, the variable f_i^{th} will be zero whenever possible, hence, we do not have to enforce this. Constraints (4.7) relate the z and y variables, which ensure that visited customers are only assigned to active operators. The rationale behind this constraints is to strengthen the formulation when $\beta = 0.0$. Constraints (4.8)-(4.10) have the same meaning as constraints (3.4)-(3.6) in formulation PS of Section 3.1.1. Finally, constraints (4.11) enforce the variables to be binary.

4.1.1.1 Improvements of AS_0

We observe that the following constraints are also valid for AS_0:

$$f_i^{th} \leq x_i^{t+h} \quad i \in I, t \in \{0, ..., |T|-1\}, h \in H_i^t \tag{4.12}$$

We denote by Ω_0 the domain defined by constraints (4.2)-(4.11) and by Ω_1 the domain defined by (4.2), (4.4)-(4.12). Let also $\overline{\Omega}_0$ and $\overline{\Omega}_1$ denote the respective domains when the binary conditions on the variables are replaced by non-negativity constraints. That is:

$\Omega_0 = \{(x, y, z, f)$: satisfying (4.2)-(4.11)$\}$.
$\Omega_1 = \{(x, y, z, f)$: satisfying (4.2),(4.4)-(4.12)$\}$.

$\overline{\Omega}_0 = \{\mathbf{0} \leq (x, y, z, f) \leq \mathbf{1}$: satisfying (4.2)-(4.10)$\}$.
$\overline{\Omega}_1 = \{\mathbf{0} \leq (x, y, z, f) \leq \mathbf{1}$: satisfying (4.2), (4.4)-(4.10),(4.12)$\}$.

where **0** and **1** are vectors of appropriate dimensions with 0's and 1's, respectively. Next we see that both Ω_0 and Ω_1 give equivalent formulations for the A-MSSP. In particular, the following statements hold:

Proposition 1. $\Omega_0 = \Omega_1$.

Proof. (a) $\Omega_1 \subseteq \Omega_0$. Consider $(x, y, z, f) \in \Omega_1$ such that $(x, y, z, f) \notin \Omega_0$. Therefore, there exist indices $i \in I$ and $t \in \{1, ..., t_i\}$ with $x_i^t > \sum_{h=1}^{t_i} x_i^{t+h}$. Then, by the definition of variables, $x_i^t = 1$ and $x_i^{t+h} = 0$, for all $h \in \{1, \ldots, t_i\}$. By (4.4) and (4.5), $\sum_{h=1}^{t_i} f_i^{th} = 1$, so $f_i^{th} = 1$ for some $h \in \{1, \ldots, t_i\}$. Therefore (4.12) is violated, contradicting $(x, y, z, f) \in \Omega_1$.

(b) $\Omega_0 \subseteq \Omega_1$. Let us suppose there exists $(x, y, z, f) \in \Omega_0$ such that $(x, y, z, f) \notin \Omega_1$. Therefore, there exist indices $\bar{t} \in \{0, ..., |T|-1\}$ and $\bar{h} \in H_i^{\bar{t}}$ with $f_i^{\bar{t}\,\bar{h}} > x_i^{\bar{t}+\bar{h}}$. Then, by the definition of variables, $f_i^{\bar{t}\,\bar{h}} = 1$ and $x_i^{\bar{t}+\bar{h}} = 0$. In addition, by (4.5), $x_i^{\bar{t}} = 1$ and $f_i^{\bar{t}h} = 0$, for all $h \in H_i^{\bar{t}}$, $h \neq \bar{h}$. We apply induction to see that $x_i^{\bar{t}+h} = 0$ for all $h \in H_i^{\bar{t}}$, $h \neq \bar{h}$. For $h = 1$, by (4.6) we have $x_i^{\bar{t}} + x_i^{\bar{t}+1} \leq f_i^{\bar{t}1} + 1$. Since $f_i^{\bar{t}1} = 0$, then $x_i^{\bar{t}+1} = 0$. Let us suppose that for a given $h < m_i^{\bar{t}}$, $h \neq \bar{h}$, it holds that $x_i^{\bar{t}+s} = 0$ for all $s \in \{1, \ldots, h\}$, and let us see that if $h+1 \neq \bar{h}$, $x_i^{\bar{t}+s} = 0$ for all $s \in \{1, \ldots, h+1\}$. By (4.6), $x_i^{\bar{t}} + x_i^{\bar{t}+(h+1)} \leq \sum_{s=1}^{(h+1)-1} x_i^{\bar{t}+s} + f_i^{\bar{t}(h+1)} + 1$. Since $f_i^{\bar{t}(h+1)} = 0$ and $x_i^{\bar{t}+s} = 0$ for all $s \in \{1, \ldots, h\}$, it holds that $x_i^{\bar{t}+(h+1)} = 0$. Therefore $x_i^{\bar{t}+h} = 0$ for all $h \in H_i^t$, and (4.3) is violated. This contradicts that $(x, y, z, f) \in \Omega_0$. □

Let,

$$(AS_1) \qquad \min \; \beta \sum_{t \in T} \sum_{k \in K} z_k^t + (1-\beta) \sum_{i \in I} \sum_{t=1}^{|T|-1} \sum_{h=1}^{m_i^t} (t_i - h) f_i^{th}$$
$$s.t. \quad (x, y, z, f) \in \Omega_1$$

As a consequence of *Proposition 1* we have:

Corollary 1. *AS_1 is a valid formulation for the A-MSSP.*

Next we compare the Linear Programming (LP) relaxations of AS_0 and AS_1.

Proposition 2. $\overline{\Omega}_1 \subseteq \overline{\Omega}_0$.

Proof. Consider $(x, y, z, f) \in \overline{\Omega}_1$. Then, for all $i \in I, t \in T, h \in H_i^t$ it holds that $f_i^{th} \leq x_i^{t+h}$. Thus, for any $t \in \{0, \ldots, T - t_i\}$, we have $\sum_{h=1}^{t_i} f_i^{th} \leq \sum_{h=1}^{t_i} x_i^{t+h}$. In addition, by (4.4), $x_i^t \leq \sum_{h=1}^{t_i} f_i^{th}$. Thus, $x_i^t \leq \sum_{h=1}^{t_i} x_i^{t+h}$, which is in fact (4.3) and therefore $(x, y, z, f) \in \overline{\Omega}_0$.

□

Remark 1. *The reverse of the above condition is not true. Table 4.1 gives an example to illustrate that $\overline{\Omega}_0 \not\subseteq \overline{\Omega}_1$. In particular, Table 4.1 displays the values of the x and f variables for the solution of the LP-relaxation of AS_0 for the instance with $|I| = 1$, $|T| = 13$, $|Q| = 3$, $|R| = 1$, $t_1 = 3$ and $\beta = 0.3$. We observe that $f_1^{0,3} > x_1^3$. In fact, $f_i^{th} > x_i^{t+h}$ for $i = 1$, $t = \{3, 6, 7, 8, 9, 10\}$ and $h = 3$. Therefore,* (4.12) *is not satisfied. This implies that a solution in $\overline{\Omega}_0$ is not necessarily contained in $\overline{\Omega}_1$.*

i	t	h	$t+h$	x_i^t	f_i^{th}	x_i^{t+h}
1	0	0	0	-	-	1
1	0	1	1	-	1/17	1/17
1	0	2	2	-	0	1/17
1	0	3	3	1	16/17	53/60
1	1	3	4	1/17	1/17	7/60
1	2	3	5	1/17	1/17	7/60
1	3	3	6	53/60	53/60	13/20
1	4	3	7	7/60	7/60	7/20
1	5	3	8	7/60	7/60	1/5
1	6	3	9	13/20	13/20	1/10
1	7	3	10	7/20	7/20	1/20
1	8	3	11	1/5	1/5	1/20
1	9	3	12	1/10	1/10	0
1	10	3	13	1/20	1/20	0
1	11	-	-	1/20	-	-

Table 4.1: x and f values for the LP relaxation solution of AS_0.

As a consequence of *Proposition 2* and *Remark 1* we have:

Corollary 2. *The LP relaxation of AS_1 is tighter than the LP relaxation of AS_0.*

Moreover, AS_1 can be further reinforced with the following constraints:

$$\sum_{h=1}^{\min\{t_i,t\}} f_i^{t-h,h} = x_i^t \quad i \in I, t \in T \tag{4.13}$$

The role of Constraints (4.13) is similar to Constraints (4.4), but forcing that if customer i is visited in period t, then t is the first visit to i after some previous visit in time period $h \geq t - t_i$. In particular, if customer i is visited in period t, then it must be previously visited in no more than $p_i^t = \min\{t_i, t\}$ periods earlier. The role of Constraints (4.13) is complementary to Constraints (4.4) (for fractional solutions).

Furthermore, it can be observed that Cconstraints (4.12) are actually dominated by Constraints (4.13). By setting $t' = t + h$, we can rewrite Constraints (4.12) as $f_i^{t'-h,h} \leq x_i^{t'}$, for all $i \in I, t \in T, h \in \{1, \ldots, p_i^t\}$. By the definition of the variables, we have that $\sum\limits_{h=1}^{p_i^t} f_i^{t-h,h} \leq 1$. In particular, $\sum\limits_{h=1}^{p_i^t} f_i^{t-h,h} = 0$, if customer i is not visited in period t. Conversely, $\sum\limits_{h=1}^{p_i^t} f_i^{t-h,h} = 1$, if customer i is indeed visited in period t. Therefore, $\sum\limits_{h=1}^{p_i^t} f_i^{t-h,h} = x_i^t$, which is in fact (4.13).

We denote by $\Omega = \{(x, y, z, f) :$ satisfying (4.2),(4.4)-(4.11), (4.13)$\}$. Let

$$
\begin{aligned}
(AS) \qquad & \min \ \beta \sum_{t \in T} \sum_{k \in K} z_k^t + (1-\beta) \sum_{i \in I} \sum_{t=0}^{|T|-1} \sum_{h=1}^{m_i^t} (t_i - h) f_i^{th} \\
& s.t. \ \ (x, y, z, f) \in \Omega
\end{aligned}
$$

Preliminary computational tests indicate that the LP relaxation of AS provides a lower bound 72% tighter than those of formulation AS_0. The same tests indicate that the computing times for formulation AS are 94% smaller than those of formulation AS_0. From now on, we will use AS as the sparse formulation for the A-MSSP.

4.1.2 A class-based sparse formulation

In this section we present the class-based formulation for the A-MSSP. The details of this formulation are similar to the class-based formulation for the P-MSSP presented in Chapter 3.1.2. We adapt some parameters:

$m_j^t =$ $\min\{u_j, |T| - t\}$: number of potential periods for visits to customers of class $j \in J$ after period $t \in \{0, ..., |T| - 1\}$.

$H_j^t =$ $\{1, \ldots, m_j^t\}$: set of potential periods for visits to customers of class $j \in J$ after period $t \in \{0, ..., |T| - 1\}$.

$p_j^t =$ $\min\{u_j, t\}$: number of potential periods for visits to customers of class $j \in J$ before period $t \in T$.

In the formulation below we use the already defined variables of Chapter 3.1.2. Moreover, in order to identify the number of periods between two consecutive visits, we define the following set of general integer variables:

For $j \in J$, $t \in \{0, \ldots, |T| - 1\}$, $h \in H_j^t$,

f_j^{th} : number of customers of frequency class j who are visited consecutively in periods t and $t + h$.

Then, the class-based sparse MILP for the A-MSSP is the following:

$$(AS^c) \quad \min \; \beta \sum_{t\in T}\sum_{k\in K} z_k^t + (1-\beta)\sum_{j\in J}\sum_{t=1}^{|T|-1}\sum_{h=1}^{m_j^t}(u_j - h) f_j^{th} \tag{4.14}$$

$$s.t. \quad \sum_{h=1}^{u_j} f_j^{0h} = w_j \qquad j \in J \tag{4.15}$$

$$x_j^t = \sum_{h=1}^{u_j} f_j^{th} \qquad j \in J, t \in \{0, ..., |T| - u_j\} \tag{4.16}$$

$$\sum_{h=1}^{m_j^t} f_j^{th} \leq x_j^t \qquad j \in J, t \in \{|T| - u_j + 1, ..., |T| - 1\} \tag{4.17}$$

$$\sum_{h=1}^{p_j^t} f_j^{t-h,h} = x_j^t \qquad j \in J, t \in T \tag{4.18}$$

$$z_k^t \leq \sum_{j\in J} y_{jk}^t \qquad k \in K, t \in T \tag{4.19}$$

$$x_j^t = \sum_{k\in K} y_{jk}^t \qquad j \in J, t \in T \tag{4.20}$$

$$\sum_{j\in J} y_{jk}^t \leq Q z_k^t \qquad k \in K, t \in T \tag{4.21}$$

$$Q z_k^t \leq \sum_{j\in J} y_{j,k-1}^t \qquad k \in K \setminus \{1\}, t \in T \tag{4.22}$$

$$z_k^t \in \{0, 1\} \qquad k \in K, t \in T \tag{4.23}$$

$$x_j^t,\, y_{jk}^t, f_j^{th} \quad \text{integer,} \qquad j \in J, k \in K, t \in T, h \in H_j^t \tag{4.24}$$

In AS^c the objective (4.14) minimizes a weighted sum of the total number of operators used in the time horizon and the total earliness. By assigning different values to $\beta \in [0, 1]$, both criteria can be considered within different scenarios. Constraints (4.15) guarantee that the first visit period for customers in class j occurs within their service interval. Constraints (4.16) and (4.17) are logical constraints, which relate the x and f variables. Constraints (4.18) are logical constraints, which force that visits to customers in class j have a previous visit in no more than p_j^t periods before. Constraints (4.19) relate the z and y variables, and ensure that visited customers are only assigned to active operators. The rationale behind these constraints is to strengthen the formulation when $\beta = 0.0$. Constraints (4.20)-(4.22) have the same meaning as constraints (3.11)-(3.13) of PS^c of Chapter 3.1.2. Finally, Constraints (4.23) enforce variables z_k^t to be binary, while Constraints (4.24) enforce integrality on the variables x_j^t, y_{jk}^t and f_j^{th}.

As can be observed, AS^c has a structure similar to that of AS of Section 4.1.1. The difference between both formulations relies on the use of the set of frequency classes (J) instead of the set of customers (I), changing the nature of variables x_j^t, y_{jk}^t and f_j^{th} to be general integer and reducing therefore the number of variables of each of these types. When $J = I$, variables x_j^t, y_{jk}^t and f_j^{th} become binary variables. In this case, AS yields a tighter LP relaxation than AS^c due to the existence of Constraints (4.6).

Chapter 5

Branch-and-Price for the A-MSSP

Branch-and-price (BP) is a solution method for combinatorial optimization, widely used for solving MILPs with many variables. BP is a branch-and-bound method in which at each node of the search tree new columns may be added to the current subproblem. At the start of the algorithm many columns are excluded from the formulation in order to reduce the computational and memory requirements. Then, columns are incorporated to current formulation as needed. The process of adding new columns is usually referred to as column generation. It is based on the idea that for large problems most columns will have their corresponding variable at value zero in an optimal solution (Barnhart et al., 1998). Indeed, such columns are unnecessary for solving the problem.

Typically, a column generation algorithm is applied to a reformulation, known as the Master Problem (MP). Since the reformulation usually contains too many variables, a restricted version is used, called the Restricted Master Problem (RMP), which only considers a subset of the total columns (Feillet, 2010). New columns to MP are generated from the optimal solution to the RMP. When MP and RMP are MILPs, as in our case, this process is applied to their LP relaxations (LMP and LRMP, respectively). The motivation for resorting to column generation in this case is that it is expected that the problems that are solved give better LP bounds than their original counterparts.

To check for optimality with respect to LMP of the optimal solution to LRMP, we solve a pricing problem to find columns to enter the basis and reduce the objective function value. This involves solving a pricing problem, which consists of finding a column with negative reduced cost. When the pricing problem is difficult to solve, heuristic and local search methods are generally used in order to find columns with negative reduced costs (Mehrotra and Trick, 2007). The subproblem is then only solved to completion in order to prove that an optimal solution to LRMP is also optimal o LMP. Each time a column with negative reduced cost is found, it is added to the RMP and LRMP is re-optimized. If no columns can enter the basis and the solution to LRMP is not integer, then branching occurs. Figure 5.1 gives a general outline of the branch-and-price algorithm.

In this chapter, we describe the BP developed to solve the A-MSSP. In the following sections we present two different formulations, both suitable for column generation. For each formulation, we provide the pricing problem necessary for the generation of columns. In addition, we include the algorithmic details for the implementation. The results of the computational experiments as well as the comparison with the results of the A-MSSP formulations of Chapter 4.1 are shown in Chapter 7.4.

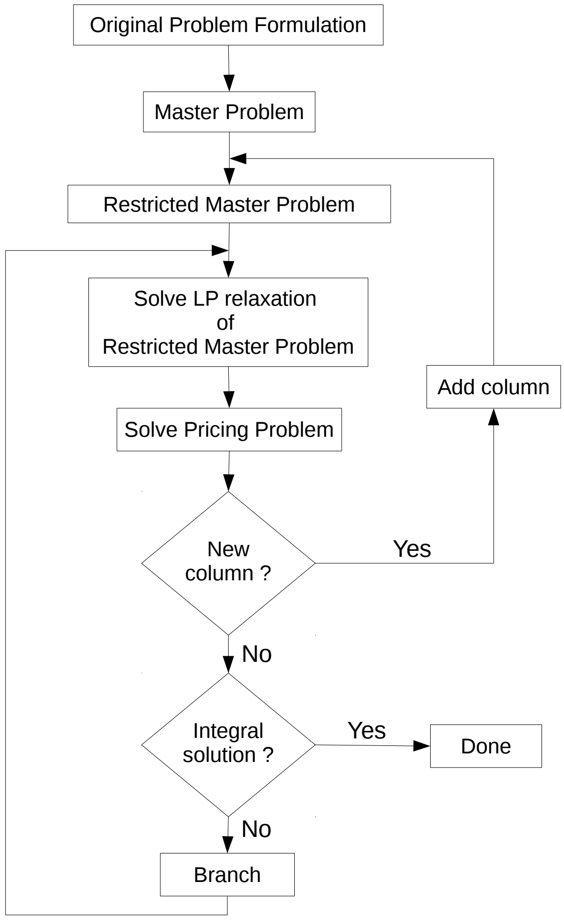


Figure 5.1: Outline of the branch-and-price algorithm

5.1 Column generation formulations for the A-MSSP

Below, we present two alternative MILP formulations for the A-MSSP, both suitable for column generation. These formulations are reformulations of the sparse MILP formulations presented in Chapter 4.1. Therefore, we refer to the first one as "customer-based" and the second one as "class-based" formulation.

5.1.1 A customer-based column generation formulation

The A-MSSP formulation we present below can be seen as the column generation reformulation of formulation AS presented in Chapter 4.1.1. In this formulation columns correspond to *patterns* for visits. A pattern $c \subseteq I$ consists of a subset of customers that are served in a given period. For every period $t \in T$, we denote by C^t the set of all patterns for possible visits in period t, whereas $C_i^t \subseteq C^t$ is the set of all patterns for period t containing customer i. Since, for every period, a pattern can contain $0, 1, \ldots, |I|$ customers, the total number of patterns for period t is thus $|C^t| = \sum_{i=0}^{|I|} \binom{|I|}{i} = \sum_{i=1}^{|I|} \frac{|I|!}{i!(|I|-i)!}$. The cost of pattern c is computed as $n_c = \left\lceil \frac{|c|}{Q} \right\rceil$. For the first column generation formulation, we define two sets of binary decision variables:

For $t \in T \cup \{0\}$, $c \in C_t$

$$x_c^t = \begin{cases} 1 & \text{if pattern } c \text{ is visited in period } t \\ 0 & \text{otherwise} \end{cases}$$

For $i \in I, t \in \{0, ..., |T| - 1\}, h \in H_i^t$

$$f_i^{th} = \begin{cases} 1 & \text{if two consecutive visits to customer } i \text{ happen in time periods } t \text{ and } t+h \\ 0 & \text{otherwise} \end{cases}$$

Therefore the MP for the customer-based column generation formulation for the A-MSSP is as follows:

$$(MP) \quad \min \beta \sum_{t \in T} \sum_{c \in C_t} n_c^t x_c^t + (1-\beta) \sum_{i \in I} \sum_{t=1}^{|T|-1} \sum_{h=1}^{m_i^t} (t_i - h) f_i^{th} \tag{5.1}$$

$$s.t. \quad \sum_{h=1}^{t_i} f_i^{0h} \geq 1 \qquad i \in I \tag{5.2}$$

$$\sum_{c \in C_i^t} x_c^t \leq 1 \qquad i \in I, t \in T \tag{5.3}$$

$$\sum_{c \in C_i^t} x_c^t = \sum_{h=1}^{t_i} f_i^{th} \qquad i \in I, t \in \{0, ..., |T| - t_i\} \tag{5.4}$$

$$\sum_{h=1}^{m_i^t} f_i^{th} \leq \sum_{c \in C_i^t} x_c^t \qquad i \in I, t \in \{|T| - t_i + 1, ..., |T| - 1\} \tag{5.5}$$

$$\sum_{h=1}^{p_i^t} f_i^{t-h,h} = \sum_{c \in C_i^t} x_c^t \qquad i \in I, t \in T \tag{5.6}$$

$$x_c^t,\ f_i^{th} \in \{0, 1\} \qquad i \in I, t \in T, c \in C_t, h \in H_i^t \tag{5.7}$$

As in formulation *AS* of Chapter 4.1.1, the weight $\beta \in [0,1]$ in the objective (5.1) allows to consider different scenarios on the number of operators and the number of early periods within visits. Constraints (5.2) guarantee that the first visit period for each customer takes place within his service interval. Constraints (5.3) avoid multiple visits to the same customer within the same period. Constraints (5.4) and (5.5) relate the x and f variables, and ensure that the number of periods between two consecutive service periods for the same customer never exceeds his service interval. Additionally, Constraints (5.6) ensure that if customer i is visited in period t, then it must be previously visited in no more than $p_i^t = \min\{t_i, t\}$ periods earlier. Finally, Constraints (5.7) enforce the variables to be binary.

Remark 2. *Because of the definition of x variables, we have that*

$$\sum_{c \in C_t} x_c^t \leq 1 \qquad t \in T \tag{5.8}$$

is an optimality cut for MP. Furthermore, it can be easily observed that Constraints (5.8) *dominate* (5.3).

Formulation MP has an exponential number of x variables and $O(n^3)$ variables f. We call Restricted Master Problem RMP to MP restricted to only a subset of the x variables but with all the f variables. We use the following notation for the dual variables associated with the LP relaxation of RMP:

γ^t: $t \in T$, for Constraints (5.8).

φ_i^t: $i \in I, t \in \{0, ..., |T| - t_i\}$, for Constraints (5.4).

π_i^t: $i \in I, t \in \{|T| - t_i + 1, ..., |T| - 1\}$, for Constraints (5.5).

σ_i^t: $i \in I, t \in T$, for Constraints (5.6).

To formulate the pricing problem for period $t \in T$ we use the following set of binary variables:

For $k \in K$,

$$v_k = \begin{cases} 1 & \text{if operator } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

For $i \in I$, $k \in K$,

$$y_{ik} = \begin{cases} 1 & \text{if customer } i \text{ is visited by operator } k \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the Pricing Problem for period t is:

$$(P^t) \;\; z^t = \gamma^t + \;\; \min \beta \sum_{k \in K} v_k - \sum_{i \in I} \alpha_i \sum_{k \in K} y_{ik} \tag{5.9}$$

$$s.t. \;\; \sum_{k \in K} y_{ik} \leq 1 \qquad i \in I \tag{5.10}$$

$$\sum_{i \in I} y_{ik} \leq Q v_k \qquad k \in K \tag{5.11}$$

$$Q v_k \leq \sum_{i \in I} y_{i,k-1} \qquad k \in K \setminus \{1\} \tag{5.12}$$

$$v_k, \, y_{ik} \in \{0, 1\} \qquad i \in I, k \in K \tag{5.13}$$

with $\alpha_i = -\varphi_i^t + \pi_i^t + \sigma_i^t$.

Objective (5.9) minimizes the reduced cost of any possible visit pattern for period t relative to the dual multipliers vector $(\gamma^t, \varphi_i^t, \pi_i^t, \, \sigma_i^t)$. Constraints (5.10) avoid multiple visits to the same customer. Constraints (5.11) are capacity constraints, which ensure that the number of customers assigned to each operator does not exceed the operator capacity. Constraints (5.12) are symmetry breaking constraints imposing that, at each period, the operator k is not used unless operators $1, \ldots, k-1$ are full, i.e., each of them has Q assigned customers. Finally, Constraints (5.13) enforce the variables to be binary.

If $z^t < 0$, we obtain a new variable, which may improve the current LP solution of the RMP. The new variable x_c^t is associated with the pattern $c = \{i \in I : y_{ik} = 1, k \in K\}$ with cost $n_c^t = \sum_{k \in K} v_k$.

To solve P^t we can apply, for every $t \in T$, a simple algorithm. Initially we sort the indices of customers with strictly positive coefficient α_i, by non increasing values of the coefficients, i.e. $\alpha_{i_1} \geq \alpha_{i_2} \geq ...\alpha_{i_s} > 0$. At each iteration we try to activate a new operator and to assign to it up to Q customers, not previously assigned. Customers are tentatively assigned to the initial ordering. If i_{k-1} denotes the index of the last customer assigned at iteration k-1, at iteration k we assign to a new operator customers $i_k, i_{k+1}...i_h$, where $h = \min\{(k-1)+Q, s\}$, provided that $\sum_{r=h}^{\min\{Qk,s\}} \alpha_{i_r} > \beta$. Otherwise the process terminates and the operators activated so far, together with their assigned customers, define an optimal solution to the pricing problem. Algorithm 5.1 gives an outline of the pricing solving algorithm for P^t.

Algorithm 5.1: Solution algorithm for P^t

Data: β, α_i, $i \in I$

sort $\alpha_{i_1} \geq \alpha_{i_2} \geq \ldots \geq \alpha_{i_s}$;
stop $\leftarrow$ false ;
$k \leftarrow 1$;
while *not stop* **do**
    $h = Q(k-1)+1$;
    if $\sum_{r=h}^{\min\{Qk,s\}} \alpha_{i_r} > \beta$ *and* $k < |K|$ **then**
        $y_{i_r,k} = 1,\ r = h, \ldots, \min\{Qk, s\}$;
        $v_k = 1$;
        $k \leftarrow k+1$;
    else
        stop;
    end
end

5.1.2 A class-based column generation formulation

In this section we present a class-based version of the column generation formulation presented in Section 5.1.1, which is a column generation version of formulation AS^c of Section 4.1.2.

In this formulation columns correspond again to patterns of visits, but referring to the number of customers of class $j \in J$ that are visited in a given period. We use the following notation in addition to that introduced in Section 4.1.2.

A pattern is represented by a vector $c = (a_1^c, \ldots, a_j^c, \ldots, a_{|J|}^c)$ whose *j-th* component, $a_j^c \leq w_j$, indicates the number of customers of class j that are visited in a given period. As before n_c is the cost of pattern c, which is now given by $n_c = \left\lceil \frac{\sum_{j=1}^{|J|} a_j^c}{Q} \right\rceil$. C^t is the set of all possible patterns c for period t. For each period $t \in T$, the number of all possible patterns is $\prod_{j=1}^{|J|}(w_j+1)$. Since each pattern can be used in any period $t \in T$, we define C as the set of all possibles patterns in the entire time horizon, $C = \bigcup_{t \in T} C^t$, and therefore $|C| = |T| \prod_{j=1}^{|J|}(w_j+1)+1$.

For the class-based column generation formulation we use the following sets of decision variables:

For $t \in T \cup \{0\}$, $c \in C^t$

$$x_c^t = \begin{cases} 1 & \text{if the pattern } c \text{ is collected at period } t \\ 0 & \text{otherwise} \end{cases}$$

For $j \in J, t \in \{0, ..., |T| - 1\}, h \in H_j^t$

f_j^{th} = the number of customers of class j that are consecutively visited in periods t and $t + h$.

The class-based Master Problem for the A-MSSP is as follows:

$$(MP^c) \min \beta \sum_{t \in T} \sum_{c \in C^t} n_c x_c^t + (1 - \beta) \sum_{j \in J} \sum_{t=1}^{|T|-1} \sum_{h=1}^{m_j^t} (u_j - h) f_j^{th} \tag{5.14}$$

$$s.t. \quad \sum_{h=1}^{u_j} f_j^{0h} = w_j \qquad j \in J \tag{5.15}$$

$$\sum_{c \in C^t} a_j^c x_c^t \leq w_j \qquad j \in J, t \in T \tag{5.16}$$

$$\sum_{c \in C^t} a_j^c x_c^t = \sum_{h=1}^{u_j} f_j^{th} \qquad j \in J, t \in \{0, ..., |T| - u_j\} \tag{5.17}$$

$$\sum_{h=1}^{m_j^t} f_j^{th} \leq \sum_{c \in C^t} a_j^c x_c^t \qquad j \in J, t \in \{|T| - u_j + 1, ..., |T| - 1\} \tag{5.18}$$

$$\sum_{h=1}^{p_j^t} f_j^{t-h,h} = \sum_{c \in C^t} a_j^c x_c^t \qquad j \in J, t \in T \tag{5.19}$$

$$x_c^t \in \{0, 1\} \qquad t \in T, c \in C^t \tag{5.20}$$

$$f_j^{th}, \ \text{integer} \qquad j \in J, t \in \{0, ..., |T| - 1\}, h \in H_j^t \tag{5.21}$$

Similarly to formulation AS^c of Chapter 4.1.2, the weight $\beta \in [0, 1]$ in the objective (5.14) allows us to consider different scenarios for the number of operators and the number of early periods within visits. Constraints (5.15) set the starting point for visits along the time horizon. Constraints (5.16) limit to w_j the number of customers of each class that can be visited in the same period. Constraints (5.17) and (5.18) relate the x and f variables, and ensure that the number of periods between two consecutive services to the same customer never exceeds his service interval. Constraints (5.19) guarantee that if customers of class j have been visited in period t, then they must be previously visited in no more than $p_i^t = \min\{v_j, t\}$ periods earlier. Finally, Constraints (5.20) enforce variables x_c^t to be binary, while Constraints (5.21) enforce the integrality on the variables f_j^{th}.

Observe that due to the definition of x variables, we have that Constraints (5.8) also define an optimality cut for the MP^c. Note also that Constraints (5.16) are dominated by (5.8).

Formulation MP^c also has an exponential number of x variables and $O(n^3)$ variables f. We call Restricted Master Problem (RMP^c) to (MP^c) restricted to only a subset of the x variables but with all the f variables. We use the following notation for the dual variables associated with the LP relaxation of RMP^c:

γ^t: $t \in T$, for Constraints (5.8).

φ_j^t: $j \in J, t \in \{0, ..., |T| - u_j\}$, for Constraints (5.17).

π_j^t: $j \in J, t \in \{|T| - u_j + 1, ..., |T| - 1\}$, for Constraints (5.18).

σ_j^t: $j \in J, t \in T$, for Constraints (5.19).

For every period $t \in T$, the Pricing Problem $\left(P_c^t\right)$ for finding "attractive" visit patterns for time period t. P_c^t uses the following decision variables:

For $k \in K$,

$$v_k = \begin{cases} 1 & \text{if operator } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

For $j \in J, k \in K$,

$$y_{jk} = \text{number of customers of class } j \text{ visited by operator } k$$

Then, the pricing problem for period t is:

$$\left(P_c^t\right) \; z_c^t = \gamma^t + \min \beta \sum_{k \in K} v_k - \sum_{j \in J} \alpha_j \sum_{k \in K} y_{jk} \tag{5.22}$$

$$\begin{aligned} s.t. \quad & \sum_{k \in K} y_{jk} \leq w_j & & j \in J & (5.23) \\ & \sum_{j \in J} y_{jk} \leq Q v_k & & K \in K & (5.24) \\ & Q v_k \leq \sum_{i \in I} y_{i,k-1} & & k \in K \setminus \{1\} & (5.25) \\ & v_k \in \{0,1\} & & k \in K & (5.26) \\ & y_{jk}, \quad \text{integer} & & j \in J, k \in K & (5.27) \end{aligned}$$

with $\alpha_j = -\varphi_j^t + \pi_j^t + \sigma_j^t$.

Objective (5.22) minimizes the reduced cost of any possible visit pattern for period t relative to the dual multipliers vector $(\gamma^t, \varphi_i^t, \pi_i^t, \sigma_i^t)$. Constraints (5.23) avoid multiple visits to customers. Constraints (5.24) are capacity constraints that ensure that the number of customers assigned to each operator does not exceed the operator capacity. Constraints (5.25) are symmetry breaking constraints imposing that at each period operator k is not used unless operators $1, \ldots, k-1$ are full, i.e., each of them has Q assigned customers. Finally, Constraints (5.26) and (5.27) enforce integrality on the variables.

If $z_c^t < 0$, we obtain a new variable, which may improve the current LP solution to the RMP^c. The new variable x_c^t is then defined by $c = (a_j^c)_{j\in J}$, where $a_j^c = \sum_{k\in K} y_{jk}$, for all $j \in J$ and its cost is thus, $n_c^t = \sum_{k\in K} v_k$.

To solve P_c^t we apply a variation of Algorithm 5.1 presented in Section 5.1.1. In this case we take into account the number of customers that have not been assigned of the last class used in the previous iteration.

5.2 Column generation algorithm for the A-MSSP

A sketch of a generic column generation algoritm is presented in Algorithm 5.2. Below we give the details for the main elements for the two BP implementations, one for each of the column generation formulations of Section 5.1. In particular, we describe how to provide an initial set of columns, the different strategies to apply branching, as well as the method to recover from unfeasibility after branching. We include a final section with a stabilization procedure in order to avoid the generation of too many columns during the pricing and, thus, speed up the algorithm.

Algorithm 5.2: Column generation algorithm for the MP

```
1  initialization;
2  stop ← false ;
3  while not stop do
4      solve LP relaxation of RMP;
5      if RMP feasible then
6          solve the pricing problem;
7          if new variables then
8              go to 4 ;
9          else
10             if integer solution then
11                 stop;
12             else
13                 do branching;
14             end
15         end
16     else
17         solve Farkas's pricing problem ;
18         go to 4 ;
19     end
20 end
```

5.2.1 Initialization

The first step of the column generation algorithm is to obtain an initial RMP. For this we need an initial set of columns (variables) which contain a feasible primal solution. Since the initial RMP determines the initial dual variables that will be used in the pricing problem, good quality initial columns are crucial for the efficiency of the algorithm. There exist several strategies to obtain these initial columns. We use a greedy procedure to obtain the feasible solution, which provides the initial feasible columns. This procedure is described in detail in Chapter 6.1.

5.2.2 Branching rules

Several branching schemes can be used to enforce the integrality of the variables of MP. General branching schemes can be found in Barnhart et al. (1998), Vanderbeck (2011). The branching rules we have applied take advantage of the specific structure of our MPs. Observe that with $v^t \in \{0, 1\}$, for all $t \in T$, the optimality cut (5.8) of Section 5.1.1 can be written as:

$$\sum_{c \in C_t} x_c^t + v^t = 1 \qquad t \in T \tag{5.28}$$

Next we present the branching rules for both MPs detailed in the previous sections.

5.2.2.1 Branching rules for formulation MP

Below we describe the three different branching strategies we have used in our BP algorithm for solving MP described in Section 5.1.1. Let, $(\overline{x}, \overline{f}, \overline{v})$ denote an optimal solution to $LRMP$. The three different branching rules are:

Strategy 1: branching on variables v^t. A first alternative is to branch on variables v^t. Branching on variable v^t forces to decide whether or not to schedule a visit in period t. In particular, we apply the most fractional variable rule, i.e., we choose to branch on variable v^{t^*} such that:

$$t^* = \arg\min_{t \in T} \left\{ \left| \overline{v}^t - 0.5 \right| \right\} \tag{5.29}$$

Figure 5.2 shows the implications for branching strategy 1.

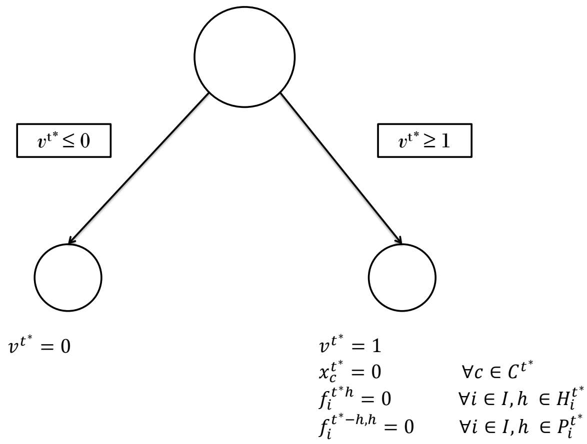


Figure 5.2: Branching strategy 1 for MP.

Strategy 2: branching on variables f_i^{th}. Since MP inherits the original variables f_i^{th} from formulation AS (Section 4.1.1), we take advantage of them and use them in one of the branching strategies. We also apply the most fractional variable branching rule. In particular, we choose to branch on variable $f_{i^*}^{t^*h^*}$ for the triplet $(i, t, h)^*$ such that:

$$(i, t, h)^* \in \arg\min_{\substack{i \in I \\ t \in T \\ h \in H_i^t}} \left\{ \left| \overline{f}_i^{th} - 0.5 \right| \right\}. \tag{5.30}$$

Figure 5.3 displays the implications of the branching strategy 2.

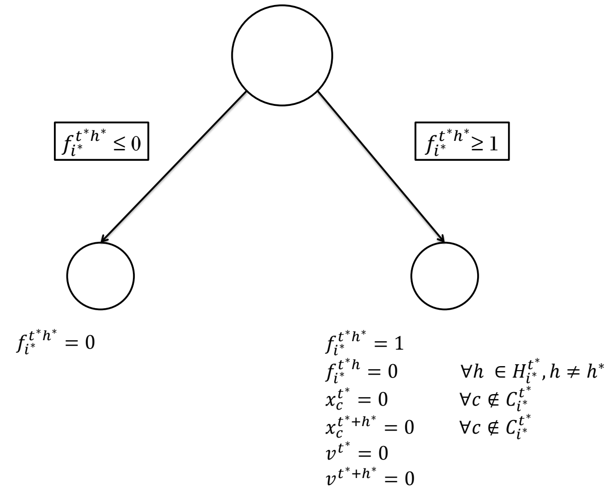


Figure 5.3: Branching strategy 2 for MP.

Strategy 3: branching on x_c^t variables. Finally, we can also branch on columns x_c^t. Following the most fractional branching rule, we choose to branch on variable $x_{c^*}^{t^*}$ for the pair $(t, c)^*$ such that:

$$(t, c)^* \in \arg\min_{\substack{t \in T \\ c \in C^t}} \left\{ \left| \overline{x}_c^t - 0.5 \right| \right\} \tag{5.31}$$

Figure 5.4 shows the implications of the branching strategy 3.

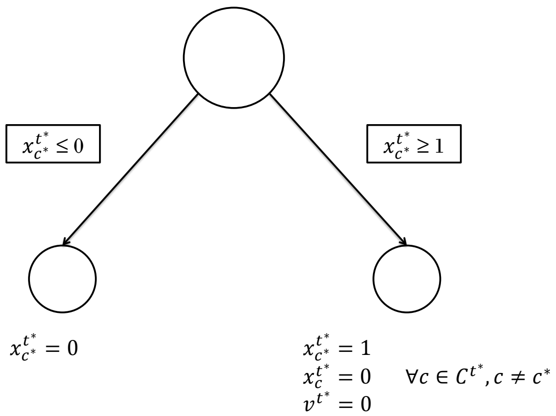


Figure 5.4: Branching strategy 3 for MP.

Branching strategies 1, 2 and 3 were applied in order to solve MP. Preliminary numerical tests showed that branching on variables v^t has a very poor performance. Furthermore, we observed that branching on variables f_i^{th} guarantees the integrality of variables x_c^t. In consequence, strategy 2 was the only branching strategy applied for solving the integer MP.

5.2.2.2 Branching rules for formulation MP^c

We next describe the branching strategies used for solving MP^c described in Section 5.1.2. Let, $(\overline{x}, \overline{f}, \overline{v})$ denote tan optimal solution to $LRMP^c$. The three different branching strategies we have used are:

Strategy 1: branching on variables v^t. Similarly to formulation MP, branching on variable v^t forces to decide whether or not to schedule customers in period t. We apply the same strategy 1 as described in Section 5.2.2.1. That is, we select variable v^{t^*} with the most fractional value, following equation (5.29). Figure 5.5 displays the implications of this strategy.

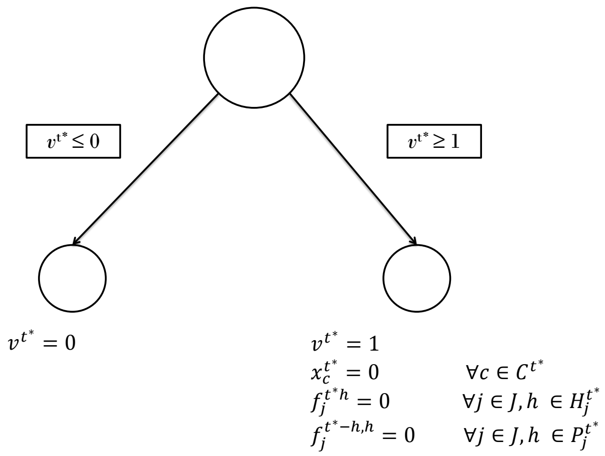


Figure 5.5: Branching strategy 1 for the MP^c.

Strategy 2: branching on variables f_j^{th}. Analogously to formulation MP, formulation MP^c inherits the variables f_j^{th} from formulation AS^c of Section 4.1.2. Following the most fractional branching rule, we choose to branch on variable $f_{j^*}^{t^*h^*}$ for the triplet $(j, t, h)^*$ such that:

$$(j, t, h)^* \in \arg \min_{\substack{j \in J \\ t \in T \\ h \in H_j^t}} \left\{ \min \left\{ \left\lceil \overline{f}_j^{th} \right\rceil - \overline{f}_j^{th}, \overline{f}_j^{th} - \left\lfloor \overline{f}_j^{th} \right\rfloor \right\} \right\} \tag{5.32}$$

Figure 5.6 shows the implications of the branching strategy 2 for formulation MP^c.

Strategy 3: branching on x_c^t variables. Similarly to strategy 3 for formulation MP, we select variable $x_{c^*}^{t^*}$ with the most fractional value by following equation (5.31). The implications of the branching strategy 3 for formulation MP^c are displayed in Figure 5.4.

Branching strategies 1, 2 and 3 were applied to solve the MP^c. In contrast to the behaviour observed with the branching strategies for the MP, preliminary numerical tests for MP^c showed that branching on variables v^t presents a good performance, specially when strategy 1 is applied together with strategies 2 and 3. Contrary to formulation MP, for formulation MP^c, branching on variables f_j^{th} does not guarantee the integrality on variables x_c^t. The best observed performance was to apply sequentially branching strategies 1, 2 and 3. That is, we explore the branching tree first branching on variables v^t, then on variables f_j^{th} and finally on variables x_c^t.

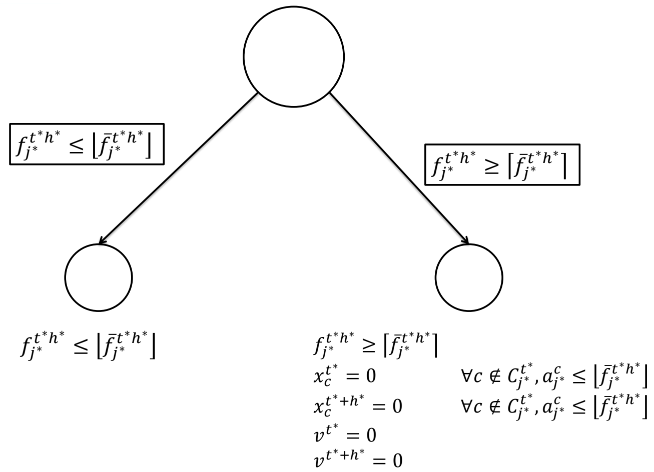


Figure 5.6: Branching strategy 2 for MP^c.

5.2.3 Farkas pricing

Infeasibility of the LRMP may arise after branching. Even if Farkas's Lemma (Farkas, 1894) can prove infeasibility, what we need algorithmically is to find a new variable to add to the MRP and destroy this infeasibility. As we see below, such a variable can be found (or concluded that it does not exists) by solving the Farkas pricing, which is, in fact, the standard pricing with cost coefficients equal to 0 and with an extreme dual ray vector instead of the dual variable vector.

For the customer-based formulation MP, the Farkas pricing problem for every period $t \in T$, is formulated as the P^t of Section 5.1.1, in which vector $(\gamma^t, \varphi_i^t, \pi_i^t, \sigma_i^t)$ defines the dual rays associated with MP's constraints (5.8),(5.4), (5.5) and (5.6), respectively. Analogously, for the class-based formulation MP^c, the Farkas pricing problem for every period $t \in T$, is formulated as the P_c^t of Section 5.1.2, with vector $(\gamma^t, \varphi_i^t, \pi_i^t, \sigma_i^t)$ as the dual rays associated with MP^c's constraints (5.8) (5.17), (5.18) and (5.19), respectively.

5.2.4 Stabilization procedure

The convergence of column generation algorithms when using simplex algorithm is usually very poor, specially for large and degenerate problems. The main reason is that the dual values obtained from the RMP in the first iterations of the procedure do not appear in the optimal dual solution, and therefore, they move from one extreme solution of the dual domain to another distant one. There are several approaches to limit this erratic behaviour of the dual variables. Most of them include the modification of the RMP and introducing several parameters that have to be adjusted, see Lübbecke (2011). In our work we apply a different approach that does not need any modification of the RMP and only incorporates a single parameter to be adjusted. This stabilization procedure was first introduced by Wentges (1997) and developed later by Pessoa et al. (2010) as the smoothing approach. The basic idea of the smoothing approach is to combine the current dual solution, λ^*, with a previous one, $\hat{\lambda}$. That is, for the pricing problem, we do not use the current value λ^* but the convex combination:

$$\lambda = \Delta\lambda^* + (1 - \Delta)\hat{\lambda} \tag{5.33}$$

with $0 \leq \Delta \leq 1$. When a promising column is found relative to vector λ, it is added to the RMP only if this column has a negative reduced cost with respect to λ^* as well. Furthermore, when the dual multipliers λ improve the dual bound, we update the best known vector to $\hat{\lambda} = \lambda$. We denote by $L(\hat{\lambda})$ the best (highest) lower bound obtained so far, and by Z_{MP} the current primal upper bound. Even when no column is added to the RMP (usually called as *misprice*), the dual bound always improves by at least a factor of $1/(1-\Delta)$. Because the performance of the smoothing stabilization approach highly depends on the selection of the parameter Δ, we have tried with several strategies for defining and updating Δ. These strategies are based on Violin (2014). In particular, the Δ updating scheme strongly depends on the relative difference between the upper and lower bounds of the MP, defined as:

$$Gap = \frac{Z_{MP} - L(\hat{\lambda})}{L(\hat{\lambda})} \tag{5.34}$$

In addition, we use a value ε to stop the stabilization procedure. The strategies we have used are the following:

Strategy 1. We set Δ equal to a fixed value Δ_{fix}. When $Gap < \varepsilon$, we stop the stabilization by setting $\Delta = 1$.

Strategy 2. We use $\Delta_{init} \in (0, 1]$ as a initial value for Δ. Every time $Gap < 1 - \Delta_{init}$ we make $\Delta = 1 - Gap$. When $Gap < \varepsilon$, then $\Delta = 1$.

For testing the different stabilization strategies, we solved the LP relaxation of MP^c using smoothing stabilization. Table 5.1 shows the summary comparison for solving the LP relaxation of the MP^c with strategies 1, 2 and without any stabilization. We use the benchmark instances as well as the computational environment described in Chapter 7.1. The table reports the total time (in seconds) and the total number of generated columns over the five instances of every size and every value of β. For strategies 1 and 2 the results are presented for the best value for Δ_{fix} and Δ_{init}, respectively, i.e, 0.1. For both strategies we set $\varepsilon = 0.01$. Results for the best strategy for each instance size are represented in bold.

We can observe that, in general, stabilization helps reducing not only the number of added columns but also computing times as well. In addition, strategy 2 outperforms strategy 1, for values of $\beta \leq 0.5$. For instances with $\beta > 0.5$, although strategy 1 produces a smaller number of columns, it requires longer computing times than those of strategy 2. Therefore, we can say that strategy 2 favors the efficiency of the column generation algorithm. On average, strategy 2 allows solutions 18.1% faster generating 12.1% fewer columns.

	Strategy		$I10$		$I30$			$I50$			Total
			$Q5$	$Q10$	$Q5$	$Q10$	$Q30$	$Q5$	$Q10$	$Q50$	
$\beta = 0.2$	No stab	Time	**48.2**	49.7	69.8	106.6	82.9	**86.4**	91.0	94.1	628.6
		Cols	2927	2410	4219	5984	**3791**	4414	5171	**4391**	33307
	Strategy 1	Time	57.1	45.1	55.6	104.3	71.3	89.8	**66.2**	**77.0**	566.4
		Cols	2943	2445	3003	5311	3964	**3550**	3854	4438	29508
	Strategy 2	Time	53.8	**41.6**	**55.1**	**93.7**	**70.6**	88.9	67.0	79.4	**550.0**
		Cols	**2860**	**2365**	**2994**	**5249**	3830	3645	**3786**	4448	**29177**
$\beta = 0.5$	No stab	Time	**44.4**	44.8	75.2	105.4	**70.8**	93.4	84.5	87.1	605.5
		Cols	2617	2058	4148	5674	3157	4573	4877	**3733**	30837
	Strategy 1	Time	48.4	36.0	**54.0**	94.0	59.8	90.2	62.2	69.5	514.0
		Cols	2525	**2032**	3013	4899	3292	**3478**	3622	3927	26788
	Strategy 2	Time	46.0	**35.9**	56.2	**89.3**	**58.8**	**83.4**	**61.7**	**69.3**	**500.5**
		Cols	**2460**	2068	**2987**	**4839**	3200	3595	**3552**	3736	**26437**
$\beta = 0.8$	No stab	Time	**41.4**	39.0	73.8	97.9	59.0	91.4	89.1	69.4	560.9
		Cols	2337	1785	3947	5470	**2779**	4073	5006	**3304**	28701
	Strategy 1	Time	44.4	32.6	55.2	88.8	50.8	76.7	**58.8**	**58.3**	465.6
		Cols	2285	1812	**2889**	**4673**	2910	**3017**	3535	3319	**24440**
	Strategy 2	Time	42.5	**30.7**	**54.5**	**81.4**	**50.6**	**75.7**	60.8	60.9	**457.0**
		Cols	**2273**	**1781**	2935	4717	2795	3257	**3476**	3349	24583
$\beta = 1.0$	No stab	Time	**23.5**	14.2	28.7	37.9	23.5	32.8	37.2	21.3	219.1
		Cols	1536	732	1798	2432	1267	1790	2136	**981**	12672
	Strategy 1	Time	24.4	12.3	20.9	**28.7**	20.5	27.4	25.3	23.1	182.7
		Cols	**1478**	**601**	1513	2380	**1019**	**1681**	**1812**	1008	**11492**
	Strategy 2	Time	26.1	**9.3**	**19.1**	29.0	**18.8**	**26.1**	**22.5**	**15.5**	**166.3**
		Cols	1543	732	**1500**	**2263**	1179	1682	1845	999	11743

Table 5.1: Summary stabilization results for MP^c.

Chapter 6

A Heuristic Algorithm for the MSSP

As an alternative to exact solution methods for the MSSP based on formulations, we have designed, developed and implemented heuristic algorithms to obtain good quality solutions in small computing times. We present in this chapter the heuristics for the P-MSSP and the A-MSSP.

Both heuristics start by building an initial solution using a greedy procedure. Afterwards, local search is applied to improve the quality of the greedy solution. To diversify the search, this procedure is also applied to a series of perturbed solutions obtained from the initial greedy solution. The results of the computational experiments as well as the comparison with those obtained by CPLEX for the MILP formulations presented in Chapters 3 and 4 are shown in Chapter 7.5.

6.1 The greedy heuristic

In the greedy heuristic we successively build a solution by selecting a yet unscheduled customer in each iteration and finding the best calendar for him. Customers are selected by non-decreasing values of their filling intervals t_i. The best calendar C_i for a selected customer $i \in I$ is then a calendar with the minimal increase in the total number of operators used over all periods. In each iteration we have a partial solution in which the calendars of a set of customers have already been determined. For $t \in T$, we denote by Q_t the number of customers that are served in period t. Operator $k = k_t^{max}$ is referred to as the *last operator* in period t and the number of customers assigned to her in period $t \in T$ is denoted by $Q_{k_t^{max}}$. Note that $Q_{k_t^{max}} = Q$ indicates that an additional operator would be needed if one more customer should be served in this period t. Periods where the last operator is completely full are called *saturated*, and the set of such periods is denoted by $S = \{t \in T : Q_{k_t^{max}} = Q\}$. A saturated period $t \in S$ with $k_t^{max} = |K|$ is called *exhausted*, because it is not possible to allocate any more customers to it. The set of exhausted periods is denoted by $E = \{t \in S : k_t^{max} = |K|\}$.

Initially, no customer has a calendar and no operator is used. Thus, we set $Q_t = k_t^{max} = Q_{k_t^{max}} = 0$, for all $t \in T$, and $S = E = \emptyset$. In iteration p, we first select an unscheduled customer i_p and then determine the best calendar $C_{i_p} \subseteq T$ for him. For this we solve a shortest path problem in an auxiliary network. In Sections 6.1.1 and 6.1.2 we show how to do that efficiently for the P-MSSP and the A-MSSP, respectively. Before the next iteration, the values Q_t, k_t^{max}, $Q_{k_t^{max}}$, and the sets S and E are updated for all $t \in T$. The heuristic terminates when calendars have been determined for all customers. Algorithm 6.1 gives an outline of the greedy procedure. Without loss of generality, we assume that the index set of customers is sorted by non decreasing values of their filling intervals, i.e., $I = \{i_1, \dots, i_{|I|}\}$ with $t_{i_1} \leq t_{i_2} \leq \cdots \leq t_{i_{|I|}}$.

Algorithm 6.1: Greedy Procedure

Data: T, Q, K, $I = \{i_i, i_2, \ldots, i_{|I|}\}$, t_i, with $t_{i_k} \leq t_{i_{k+1}}$.

set $Q_t = k_t^{max} = Q_{k_t^{max}} = 0$, $t \in T$;
set $S = E = \emptyset$;
for $i_p \in I$ **do**
  determine the best calendar C_{i_p} for customer i_p;
  set $Q_t = Q_t + 1$, $t \in C_{i_p}$;
  set $k_t^{max} = \left\lceil \frac{Q_t}{Q} \right\rceil$, $t \in C_{i_p}$;
  set $Q_{k_t^{max}} = Q_t - Q \cdot (k_t^{max} - 1)$, $t \in C_{i_p}$;
  set $S = (S \cup \{t \in C_{i_p} : Q_{k_t^{max}} = Q\}) \setminus (S \cap C_{i_p})$;
  set $E = E \cup \{t \in C_{i_p} : t \in S, k_t^{max} = |K|\}$;
end

Next, we show how to efficiently determine the best calendar for a customer by formulating it as a shortest path problem in an auxiliary network. Hereby, we have to distinguish between the periodic and aperiodic policy.

6.1.1 Auxiliary shortest path problem for the P-MSSP

Next, we formulate the search for the best calendar for a selected customer $i = i_p$ in iteration p. To that end we define an auxiliary network $N = (V, A)$ as follows. Let $V = \{v_t : t \in T \setminus E\} \cup \{v_0, v_{|T|+1}\}$. V contains a node associated with each non-exhausted period t, plus two pseudo nodes, v_0 and $v_{|T|+1}$. Moreover, A contains three types of arcs: (a) (v_0, v_t), with $t \in \{1, \ldots, t_i\} \setminus E$; (b) $(v_t, v_{t'})$ with $t' = t + t_i$, t', $t \in T \setminus E$; and, (c) $\left(v_t, v_{|T|+1}\right)$ with $t \in T \setminus E$, $t > |T| - t_i$. Figure 6.1 visualizes the network. A node $v_t \in V$ with $t \in S$ is called saturated. Then, we define the following costs associated with the arcs of A for types (a) and (b):

$$c(v_t, v_{t'}) = \begin{cases} 1 & \text{if node } v_{t'} \text{ is saturated;} \\ \frac{Q_{k_t^{max}}}{|T|\,Q} & \text{otherwise}\,. \end{cases} \tag{6.1}$$

The rationale behind the costs for arcs entering unsaturated vertices is that we want to favor calendars where operators are well utilized. The costs are chosen such that they are dominated by the cost of adding a new operator, i.e., the overall utilization of operators is less important than the number of additional operators. For arcs of type (c), $c(v_t, v_{|T|+1}) = 0$. Any path from v_0 to $v_{|T|+1}$ in the above network corresponds to a feasible calendar for customer i: $C_i = \{t \in T \setminus E : v_t \text{ is in the path from } v_0 \text{ to } v_{|T|+1}\}$. And a shortest path yields a best calendar for customer i, i.e., one with the smallest increase in the objective function value.

6.1.2 Auxiliary shortest path problem for the A-MSSP

The set of nodes of the auxiliary network for the aperiodic service policy is identical to that for the periodic policy. Concerning the arcs, A now contains three types of arcs: (a) (v_0, v_t), with $t \in \{1, \ldots, t_i\} \setminus E$; (b) $(v_t, v_{t'})$ with $t, t' \in T \setminus E$ where $t' = t + h$, $h \in \{1, \ldots, t_i\}$; and, (c) $(v_t, v_{|T|+1})$ with $t \in T \setminus E$, $t > |T| - t_i$. Figure 6.2 visualizes the network. For the costs associated with the arcs of A, we now also need to calculate the total earliness of a calendar for

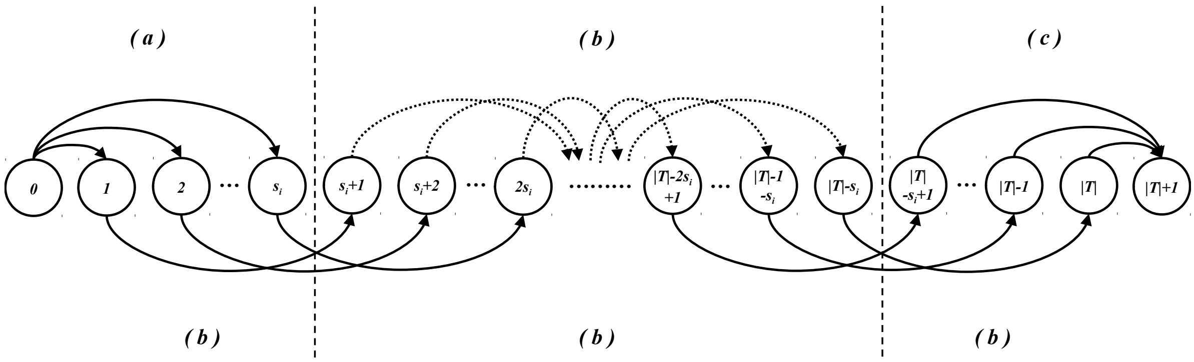


Figure 6.1: Shortest path network representation for customer $i \in I$ under the PS policy.

customer i. Thus, we define the following costs associated with the arcs of A for type (a):

$$c(v_0, v_t) \; = \begin{cases} \beta & \text{if node } v_t \text{ is saturated;} \\ \beta \frac{Q_{k_t^{max}}}{|T|Q} & \text{otherwise}\,. \end{cases} \tag{6.2}$$

The rationale behind the costs for arcs entering unsaturated vertices is the same as for the periodic policy; we just have to ensure that these costs are now also dominated by the costs of earliness. For arcs of type (b), the cost of an arc $(v_t, v_{t'})$ must now take into account not only if the destination node t' is saturated, but also the earliness between periods t and t', which is $t_i - (t' - t)$. Thus,

$$c(v_t, v_{t'}) \; = \begin{cases} (1-\beta)\,[t_i - t' + t] + \beta & \text{if node } v_{t'} \text{ is saturated;} \\ (1-\beta)\,[t_i - t' + t] + \beta \frac{Q_{k_t^{max}}}{|T|Q} & \text{otherwise}\,. \end{cases} \tag{6.3}$$

For arcs of type (c), $c(v_t, v_{|T|+1}) = 0$. Again, any path from v_0 to $v_{|T|+1}$ in the above network yields a calendar for customer i: $C_i = \{t \in T : v_t \text{ is in the path from } v_0 \text{ to } v_{|T|+1}\}$. The cost of a path reflects the increment in the objective function value of the current partial solution when incorporating calendar C_i, ignoring the utilization term $\beta \frac{Q_{k_t^{max}}}{|T|Q}$. Indeed this increment takes into account not only the number of additional operators used over all periods but also the earliness of the new partial solution. A shortest path again yields a calendar C_i for customer i with the smallest increase in costs.

6.2 Local search

In the local search we try to improve the solution obtained by the greedy heuristic. To that end, two different neighbourhoods are explored. The first one considers changing the calendar of a single customer. The second one contemplates simultaneous changes in the calendars of a pair of customers. For both policies, the goal is to change the calendars of customers in such a way that we either decrease the number of operators used over all periods (and/or the earliness for the AS policy) or, if this is not possible, the utilization of the least utilized operator in a period. The motivation for the latter is that it might be possible to get rid of this operator in subsequent iterations of the local search.

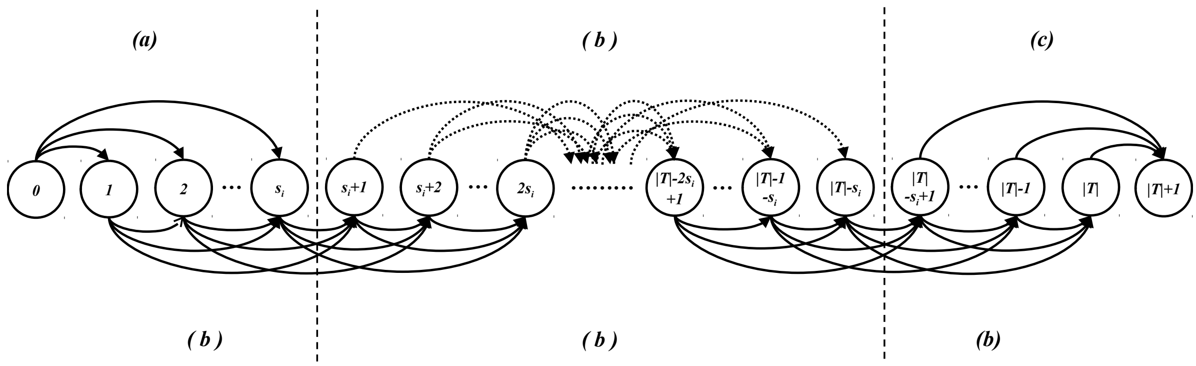


Figure 6.2: Shortest path network representation for customer $i \in I$ under the AS policy.

6.2.1 Neighborhoods

The *single-customer neighbourhood*, or simply N_1, explores solutions in which the calendars of all customers remain unchanged, except for a fixed customer $i \in I$. For this customer we want to find out if the solution would improve by using a different calendar. To that end, we first delete his calender C_i and then compute the sets S and E, and the values Q_t, k_t^{max}, and $Q_{k_t^{max}}$ for all $t \in T$. Afterwards, we determine the best calendar for i using the same approach as in Section 6.1.

Since considering the change of the calendar of a single customer may be short-sighted, we also take into account the simultaneous change of the calendar of pairs of customers. The *two-customers neighbourhood*, or simply N_2, explores solutions in which all calendars remain unchanged, except for a pair of customers $i, j \in I$. In this case we want to find out if the solution would improve if the calendars of customers i and j were different.

For the PS policy, this requires to identify the first service periods of customers i and j, and then to change all their subsequent service periods accordingly. Similarly to N_1, we can formulate the simultaneous search for the best calendars for i and j as a shortest path problem. To that end, we define the auxiliary directed network $N = (V \times V, A)$, where $V = \{v_t : t \in T \setminus E\} \cup \{v_0, v_{|T|+1}\}$ is defined as in Section 6.1.1. Nodes $(v_t, v_r) \in V \times V$ are of three types: (V_1) t and r are equal to 0; (V_2) $t, r \in T \setminus E$; (V_3) at least one of t and r is equal to $|T| + 1$ and $t > |T| - t_i$ or $r > |T| - t_j$. Moreover, arcs $\{(v_t, v_r), (v_{t'}, v_{r'})\} \in A$ are of three types: (A_1) (v_t, v_r) is a node of type (V_1) and $(v_{t'}, v_{r'})$ is a node of type (V_2), with $t' \in \{1, \ldots, t_i\}$, $r' \in \{1, \ldots, t_j\}$; (A_2) both nodes are of type (V_2) with $t' = t + t_i$ and $r' = r + t_j$; and (A_3) (v_t, v_r) is a node of type (V_2) and $(v_{t'}, v_{r'})$ is a node of type (V_3). The costs of the arcs are then defined accordingly and a shortest path through this network from node (v_0, v_0) to $(v_{|T|+1}, v_{|T|+1})$ yields the best calenders for customers i and j.

It is important to say, however, that the move for the AS policy is much more involved because earliness is allowed. As a result, we can no longer re-formulate the simultaneous search for the best calendars of i and j as a shortest path problem. Thus, we considered the two-customers neighbourhood only for the PS policy.

6.2.2 Search strategies

The neighbourhoods described in the previous section have been used within several strategies which differ from each other on the criterion to select the customers. The strategies that we use to explore neighbourhood N_1 are the following:

S_A: explores N_1 for every customer $i \in I$. The sequence in which a customer is selected is random.

S_B: first detects the operator with the minimum number of assigned customers. Then, N_1 is explored for every customer collected by this operator. If there exists more than one customer, they are selected by increasing index.

We use one single strategy to explore neighbourhood N_2, which is the following:

S_C: explores N_2 for every pair of customers $i, j \in I$. The sequence in which the customers are selected is random.

We also apply the following combinations of single strategies:

S_{AB}: strategies S_A and S_B are sequentially applied.

S_{AC}: strategies S_A and S_C are sequentially applied.

6.2.3 Diversification

In order to avoid getting trapped in local optima, we apply the local search not only to the solution produced by the greedy procedure but also to a set of perturbed solutions. The perturbed solutions are obtained by a *destroy & repair* procedure applied to the initial greedy solution, which modifies the calendar of a randomly selected subset of customers, $S \subseteq I$, with $|S| \sim U(2, |I|)$. The *destroy & repair* procedure works as follows: First, we remove the calendar of every customer in S. Then, we repair the resulting partial solution. For this we obtain, for each customer $i \in S$, a feasible calendar by solving the auxiliary shortest path problem described in Section 6.1. The order in which customers in S are selected for repairing their calendars is random.

6.2.4 Stopping criteria

We stop the local search procedure for every perturbed solution when no improvement has been found in a complete sequence of the search strategy.

Algorithm 6.2 gives the general outline of the heuristic previously described. The input $MaxDiv$ is the number of diversification iterations, whereas $F(\cdot)$ denotes the value of the objective function for a given solution. Initially, the greedy procedure is applied and then local search followed by the destroy&repair procedure are applied for $MaxDiv$ iterations.

Algorithm 6.2: Heuristic Algorithm

```
Data: MaxDiv
GreedySol ← Greedy;
BestSol ← GreedySol;
sol ← GreedySol;
iter ← 0 ;
while iter ≤ MaxDiv do
    LocalSol ← LocalSearch(sol);
    if F(LocalSol) ≤ F(BestSol) then
        BestSol ← LocalSol;
    end
    sol ← Destroy&Repair(GreedySol);
    iter ← iter + 1 ;
end
```

Chapter 7

Computational experiments

We dedicate this chapter to present and analyze the results of the computational experiments we have run. First, we give the details of the benchmark instances we have used. Then we present the numerical results and for the P-MSSP and A-MSSP formulations presented in Chapters 3 and 4. First we give the results and compare the different formulations for the periodic policy and then we do a similar analysis for the aperiodic policy. In addition, we present the computational experience of the branch-and-price of Chapter 5. The chapter concludes with the results of the heuristic algorithm presented in Chapter 6.

7.1 Data generation

Since we are not aware of any benchmark instances for the MSSP, we have generated a set of 450 benchmark instances, which we have used in all the computational experiments described in the thesis. In this section we describe the characteristics of the instances we have generated. For the number of customers we chose $|I| \in \{10, 30, 50\}$. The respective instances are labeled as "I10", "I30", and "I50". The number of periods is related to a time horizon of one month, i.e., $|T| = 30$. For the possible service intervals t_i of the customers, we considered two different settings. The first one relates the service intervals to the number of visits per month: $t_i \in \{4, 7, 15\}$, i.e., eight times, four times, and twice per month, respectively. The second one considers the intervals $t_i \in \{4, 5, \dots, 15\}$. We abbreviate the two settings by D and U, respectively. In view of the "physical" services (capacity of the bins for the WEEE collection) mentioned in Chapter 1.4, the capacity Q of the operators is 5 or 10. Moreover, to reflect the possibility of "virtual" services, we also consider the uncapacitated version of the problem, i.e., $Q = |I|$. Finally, the number of operators $|K|$ is chosen such that all problems are feasible (see Property 2 of Chapter 2.4). For this we set $|K| = \left\lceil \frac{|I|}{Q} \right\rceil$.

For each combination of values for $|I|$ and Q and for the two different settings for the service intervals, we generated five different problem instances by randomly determining service intervals for the customers according to a discrete uniform distribution over the respective set of service intervals. The resulting instances are denoted as "{D,U}_I<|I|>_Q<Q>_C<#instance>". For example, the first instance with 10 customers, an operator capacity of 5, and service intervals taken from $\{4, 7, 15\}$ is denoted "D_I10_Q5_C1".

For the AS policy we consider four different values for $\beta = \{0.2, 0.5, 0.8, 1.0\}$. The extreme value of $\beta = 1.0$ was used to analyze the effect of minimizing the total number of operators used over all periods without penalizing earliness. The opposite situation, that is, considering the minimization of total earliness without penalizing the number of operators, was originally tough to be included. However, because of Property 4 of Chapter 2.4, this scenario was discarded. The above yields a total of 400 instances, 80 for the PS policy and 320 for the AS policy.

All the computational experiments were coded in C++ and run on an Intel iCore 7@3.4 GHz with 8 GB Ram. Experiments of Chapters 3, 4 and 6 were implemented under operating system Windows 7, 64 bit. Moreover, experiments of Chapters 3 and 4 were solved using IBM ILOG CPLEX 12.5. Experiments of Chapter 5 were implemented under a platform hosting Linux 64 bit Ubuntu 12.04 operating system and solved using SCIP 3.1.1. All experiments were run with a CPU time limit of one hour.

7.2 Comparison of formulations for the P-MSSP

7.2.1 Comparison of sparse formulations

In order to observe the performance of the P-MSSP formulations introduced in Chapters 3.1.1 and 3.1.2, we present the results of a computational comparison of PS and PS^c. We use the benchmark instances and the computational environment described in Section 7.1.

Table 7.1 presents a summary of the numerical results of PS and PS^c, and each type of instances.Columns labeled z display the total number of operators over the five instances. Columns labeled *Gap* show the average percentage relative deviations of the best-known solutions with respect to lower bounds at termination; that is, $Gap = 100\frac{UB-LB}{LB}$. The average CPU times in seconds and the number of optimally solved instances are given in columns labeled *Time* and *Opt*, respectively. The detailed results can be found in Tables A.1 and A.2 in the Appendix A.

CPLEX was able to find at least one feasible solution for all instances with both formulations within the time limit of one hour. It can be observed that PS is totally outperformed by PS^c, independently of the type of instances. Formulation PS presents not only a smaller number of optimally solved instances but also larger computing times. In particular, PS is able to solve only 85% and 92.5% of the D- and U-instances, respectively. Specifically, PS is not able to guarantee optimality for instances with a larger number of customers ($I = 50$) and smaller operator capacities, $Q = 10$ and $Q = 5$. In contrast, formulation PS^c, is able to optimally solve to optimality 100% and 95% of the D- and U-instances, respectively. Only for the U-instances, with $I = 50$ and $Q = 5$, PS^c fails for guarantee optimal solutions. Furthermore, the computing times for PS^c are considerably shorter than those of PS. In particular, on average, PS^c performs 80% faster than PS.

7.2.2 Comparison of dense formulations

In order to analyze the performance of the formulations introduced in Chapters 3.2.1 and 3.2.2, we present the results from a computational experience with PD and PD^c. As in Section 7.2.1, we use the benchmark instances as well as the computational environment described in Section 7.1.

Table 7.2 presents a summary of the results with PD and PD^c, and for each type of instances.The meaning of the headings of the columns is the same as explained in Section 7.2.1. The detailed results can be found in Tables A.1 and A.2 in the Appendix A.

CPLEX was able to obtain optimal solutions for all instances with both formulations within the time limit of one hour. Furthermore, PD is totally outperformed by PS^c, independently

Instance		*PS*				PS^c			
		z	*Gap*	*Time*	*Opt*	z	*Gap*	*Time*	*Opt*
D-instances									
$I10$	$Q5$	51	0.0	0.8	5	51	0.0	0.2	5
	$Q10$	40	0.0	0.1	5	40	0.0	0.1	5
	$Q5$	140	0.0	1567.6	5	140	0.0	11.2	5
$I30$	$Q10$	79	0.0	19.2	5	79	0.0	0.6	5
	$Q30$	50	0.0	0.1	5	50	0.0	0.1	5
	$Q5$	243	4.4	3601.9	0	242	0.0	190.8	5
$I50$	$Q10$	132	2.0	1383.9	4	132	0.0	4.8	5
	$Q50$	50	0.0	0.1	5	50	0.0	0.1	5
U-instances									
$I10$	$Q5$	55	0.0	0.5	5	55	0.0	0.7	5
	$Q10$	55	0.0	0.2	5	55	0.0	0.2	5
	$Q5$	109	0.0	324.4	5	109	0.0	41.0	5
$I30$	$Q10$	71	0.0	11.3	5	71	0.0	1.8	5
	$Q30$	70	0.0	0.2	5	70	0.0	0.2	5
	$Q5$	186	1.7	2355.2	2	185	1.1	1471.8	3
$I50$	$Q10$	99	0.0	362.8	5	99	0.0	26.9	5
	$Q50$	75	0.0	0.4	5	75	0.0	0.2	5
Cumulative		1505	8.1	9628.7	71	1503	1.1	1750.7	78

Table 7.1: Summary CPLEX results for sparse formulations PS and PS^c.

of the type of instances. In particular, on average, PD^c performs almost 90% faster than PD. The main differences in computing times appear, however, on instances with a larger number of customers ($I = 50$) and smaller operator capacities ($Q = 10$ and $Q = 5$). For the rest of the instances, both formulations obtain optimal solutions in very small computing times, i.e., in less than 3 seconds.

7.2.3 Comparison between sparse and dense formulations

In Figure 7.1, we give the performance profile for all the P-MSSP formulations we presented in Chapters 3.1 and 3.2. This figure displays for each P-MSSP formulation, the percentage of optimal solutions obtained over the first 100 seconds. As mentioned, in general, dense formulations PD and PD^c outperformed sparse formulations PS and PS^c. Moreover, PS and PD are systematically outperformed by their class-based versions PS^c and PD^c, respectively. Nonetheless, we observe that the superiority of class-based formulations is more evident for D-instances than for the U-instances. As can be seen, the best performance among all P-MSSP formulations is PD^c, independently of the type of instances. All the above suggests that class-based formulations are preferred over customer-based formulations and dense formulations are preferred over sparse formulations.

The comparison between the class-based sparse formulation PS^c and the customer-based dense formulation PD deserves a special attention. Although in Figure 7.1 we observe a clear superiority of formulation PD over formulation PS^c, In Tables 7.1 and 7.2 we observe that for the D-instances with $I = 50$ and $Q = 5$, computing times for formulation PS^c are noticeable lower than those for formulation PD. The above can be easily observed in Figure 7.2. Note that this

Instance		PD				PD^c			
		z	*Gap*	*Time*	*Opt*	*z*	*Gap*	*Time*	*Opt*
D-instances									
$I10$	$Q5$	51	0.0	0.1	5	51	0.0	0.1	5
	$Q10$	40	0.0	0.2	5	40	0.0	0.1	5
	$Q5$	140	0.0	3.1	5	140	0.0	1.4	5
$I30$	$Q10$	79	0.0	0.2	5	79	0.0	0.1	5
	$Q30$	50	0.0	0.2	5	50	0.0	0.1	5
	$Q5$	242	0.0	367.2	5	242	0.0	15.0	5
$I50$	$Q10$	132	0.0	2.1	5	132	0.0	0.7	5
	$Q50$	50	0.0	0.2	5	50	0.0	0.1	5
U-instances									
$I10$	$Q5$	55	0.0	0.1	5	55	0.0	0.1	5
	$Q10$	55	0.0	0.1	5	55	0.0	0.1	5
	$Q5$	109	0.0	2.3	5	109	0.0	2.1	5
$I30$	$Q10$	71	0.0	0.2	5	71	0.0	0.2	5
	$Q30$	70	0.0	0.3	5	70	0.0	0.2	5
	$Q5$	183	0.0	24.8	5	183	0.0	12.0	5
$I50$	$Q10$	99	0.0	2.5	5	99	0.0	1.2	5
	$Q50$	75	0.0	0.4	5	75	0.0	0.2	5
Cumulative		1501	0.0	404.1	80	1501	0.00	33.9	80

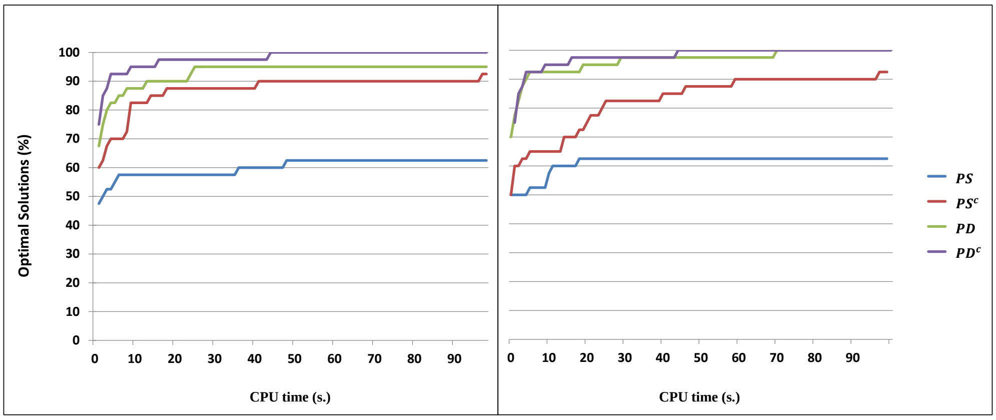
Table 7.2: Summary CPLEX results for dense formulations PD and PD^c.

behaviour cannot be observed for the U-instances. These results may suggest that the preference between formulations PS^c and PD is not only related to the type of instances (or rather, on the variety of the service intervals of the customers) but also with the size (number of customers) and the capacity of the operators. That is, formulation PS^c is preferred over formulation PD for instances with not only lower number of classes but also with a higher number of customers and smaller capacities.

7.3 Comparison of formulations for the A-MSSP

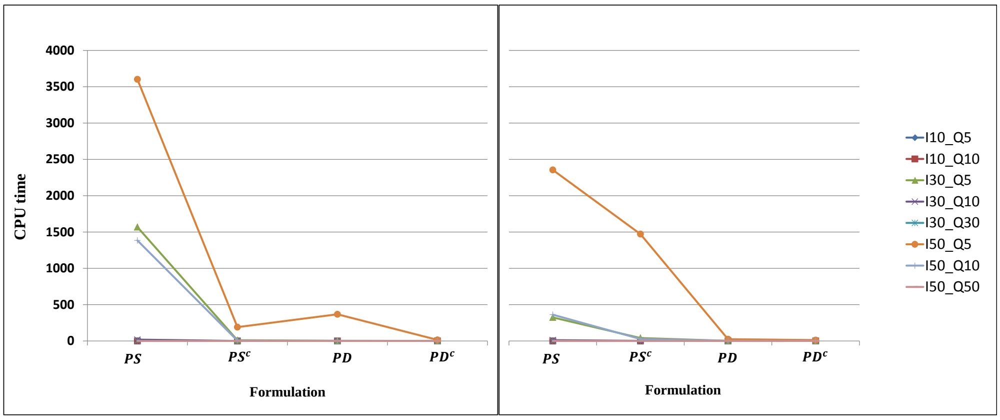
In order to observe the performance of the sparse formulations introduced in Chapter 4.1, we present the numerical results of the computational experiments for each of the MILP formulations. We use the benchmark instances and the computational environment described in Section 7.1.

Tables 7.3 and 7.4 give a summary of the results obtained with CPLEX for AS and AS^c. The results are presented for every value of β and both types of instances, D- and U-instances. Each table displays the total number of operators used (z) and the total earliness (e) of the best solutions found over the five instances. For every instance, the value of the best solution obtained with CPLEX is computed as $UB = \beta z + (1 - \beta)e$. The meaning headings of the columns have the same meaning as explained before. The detailed results can be found in Tables B.1.1, B.1.2 in the Appendix B.1 and and Tables B.2.1 and B.2.2 in the Appendix B.2, for formulations AS and AS^c, respectively.



a) D-instances b) U-instances

Figure 7.1: CPU time performance profile for the P-MSSP formulations.



a) D-instances b) U-instances

Figure 7.2: Average CPU times for the P-MSSP formulations.

As can be seen, CPLEX was able to find at least one feasible solution for all instances within the time limit of one hour for both formulations. In general, AS^c outperforms AS. For both types of instances, AS provides not only the largest number of optimally solved instances but also the smallest computing times. AS is able to guarantee optimality for 74.38% and 88.13% of the D- and U-instances respectively. The number of optimal solutions obtained with this formulation depends on the selection of the parameter β. For $\beta = 0.2$, optimality of the best solution found was proven for 77.5% and 92.5% of the D-instances and U-instances, respectively. As the value of β increases, the harder it becomes to obtain proven optimal solutions, especially for the D-instances. The lowest percentage of optimally solved instances occurs for $\beta = 0.8$, for which optimality was proven for only 60% and 65% of the D-instances and U-instances, respectively. However, the percentage of optimally solved instances for $\beta = 1.0$ is higher than for $\beta = 0.8$. In fact, the percentage of optimally solved instances for $\beta = 1.0$ is the highest for all values of β. The computing times reinforce this situation. The average values for the computing time also depends on the selection of the parameter β. As the value of β increases, the higher the CPU time. The highest computing values correspond to $\beta = 0.8$. The computing times for $\beta = 1.0$ are however, substantially smaller than for $\beta = 0.8$. The fact that $\beta = 1.0$ does neither yield the lowest percentage of optimally solved instances nor the highest average of computing time might be due to the fact that we do not penalize earliness in the objective function for

Instance		$\beta = 0.2$					$\beta = 0.5$					$\beta = 0.8$					$\beta = 1.0$				
		z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt
D-instances																					
$I10$	$Q5$	51	0	0.0	0.7	5	51	0	0.0	1.3	5	44	14	0.0	3.4	5	43	126	0.0	10.3	5
	$Q10$	40	0	0.0	0.3	5	40	0	0.0	0.4	5	37	8	0.0	1.3	5	29	342	0.0	0.8	5
	$Q5$	141	0	2.6	2612.4	3	139	1	6.5	3217.1	1	132	19	6.5	3603.4	0	130	365	0.0	97.0	5
$I30$	$Q10$	79	0	0.0	57.5	5	79	0	0.0	232.0	5	75	10	2.4	1333.8	4	70	455	4.8	2566.4	2
	$Q30$	50	0	0.0	0.9	5	50	0	0.0	2.2	5	50	0	0.0	8.5	5	35	1435	0.0	4.7	5
	$Q5$	243	0	4.4	3601.7	0	243	0	4.4	3603.1	0	236	14	3.0	3602.0	0	234	570	0.0	214.8	5
$I50$	$Q10$	132	0	4.5	2605.8	3	130	1	9.5	3261.9	1	125	10	8.3	3602.6	0	119	623	0.0	144.3	5
	$Q50$	50	0	0.0	1.6	5	50	0	0.0	2.7	5	50	0	0.0	16.7	5	35	2190	0.0	8.9	5
Cumulative		786	0	11.6	8881.0	31	782	2	20.4	10320.8	27	749	75	20.2	12171.7	24	695	6106	4.8	3047.2	37
U-instances																					
$I10$	$Q5$	55	0	0.0	1.3	5	55	0	0.0	3.3	5	46	20	0.0	20.3	5	39	197	0.0	5.5	5
	$Q10$	55	0	0.0	0.6	5	55	0	0.0	1.5	5	43	24	0.0	6.0	5	30	320	0.0	1.1	5
	$Q5$	109	0	0.0	678.8	5	109	0	0.0	1109.1	5	107	3	3.7	3604.0	0	106	421	0.0	32.3	5
$I30$	$Q10$	71	0	0.0	50.2	5	71	0	0.0	260.8	5	63	17	0.0	1070.4	5	55	413	0.0	564.6	5
	$Q30$	70	0	0.0	4.0	5	69	1	0.0	50.8	5	60	23	0.0	410.1	5	34	1532	0.0	8.1	5
	$Q5$	186	0	1.7	2627.7	2	185	0	1.1	2420.5	3	185	1	2.1	3601.7	0	183	446	0.0	142.9	5
$I50$	$Q10$	99	0	0.0	1084.0	5	99	0	0.0	2203.7	5	98	2	6.0	3411.0	1	92	608	0.0	659.5	5
	$Q50$	75	0	0.0	10.3	5	75	0	0.0	72.8	5	68	21	0.0	1468.7	5	35	2091	0.0	19.8	5
Cumulative		720	0	1.7	4457.0	37	718	1	1.1	6122.6	38	670	111	11.8	13592.2	26	574	6028	0.0	1433.8	40

Table 7.3: Summary CPLEX results for formulation AS.

$\beta = 1.0$, in contrast to $\beta = 0.8$. The optimality gap also underlines the difficulty to solve the problem with AS. The gap values again depend on the selection of the parameter β and present a behaviour similar to the percentage of optimally solved instances and the computing times. In general, the average gaps increase as the value of β increases, with the exception of $\beta = 1.0$, which has the lowest average gaps. The hardness of the problem also depends on the size of the instances and the value of the Q. Instances of "I10" are optimally solved for every value of β and Q. For instances of "I30", optimality becomes more difficult to prove for larger values of β and smaller values of Q. For the "I50" instances, optimality is almost unreachable already for values of $Q < |I|$, specially for D-instances. In general, the smaller the value of Q and the larger value of β, the more difficult to solve the problem. In general, D-instances are more difficult to solve than U-instances. This suggests that, for AS, the difficulty to solve the problem increases with the number of different of service intervals.

AS^c, obtains optimal solutions for 99.25% and 96.25% of the D- and U-instances, respectively. In particular, AS^c guarantees optimality for all sizes of instances and every value of Q and β. The only exception is for "I50" instances with $Q = 5$. Similarly to AS, the average values for the computing times depend on the selection of the parameter β. As the value of β increases, the higher the CPU times. The highest computing times again correspond to $\beta = 0.8$. Furthermore, the computing times for $\beta = 1.0$ are also substantially lower than for $\beta = 0.8$. This behaviour coincides with the results of AS, and is due to the fact that, in contrast to $\beta = 0.8$, for $\beta = 1.0$, no formulation penalizes earliness in the objective function. Since AS^c is able to solve 100% of the instances, the optimality gaps are basically null and they do not present the same behaviour as for AS. Nevertheless, the highest values for the optimality gaps are again for $\beta = 0.8$. For a visual comparison of the behaviour of AS and AS^c, Figures 7.3 and 7.4 display the average

Instance		$\beta = 0.2$					$\beta = 0.5$					$\beta = 0.8$					$\beta = 1.0$				
		z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt	z	e	Gap	$Time$	Opt
D-instances																					
$I10$	$Q5$	51	0	0.0	0.4	5	51	0	0.0	0.3	5	44	14	0.0	0.8	5	43	135	0.0	0.8	5
	$Q10$	40	0	0.0	0.2	5	40	0	0.0	0.3	5	37	8	0.0	0.6	5	29	219	0.0	0.2	5
	$Q5$	140	0	0.0	17.2	5	139	1	0.0	24.8	5	133	15	0.0	136.3	5	130	321	0.0	0.8	5
$I30$	$Q10$	79	0	0.0	1.5	5	79	0	0.0	1.6	5	75	10	0.0	4.2	5	70	419	0.0	3.5	5
	$Q30$	50	0	0.0	0.2	5	50	0	0.0	0.2	5	50	0	0.0	0.2	5	35	992	0.0	0.2	5
	$Q5$	242	0	0.0	733.9	5	242	0	0.0	786.2	5	236	12	0.2	1785.5	4	234	529	0.0	1.4	5
$I50$	$Q10$	132	0	0.0	8.2	5	130	1	0.0	13.3	5	125	10	0.0	21.1	5	119	485	0.0	0.9	5
	$Q50$	50	0	0.0	0.2	5	50	0	0.0	0.2	5	50	0	0.0	0.4	5	35	1565	0.0	0.2	5
Cumulative		784	0	0.0	761.8	40	781	2	0.0	827.0	40	750	69	0.2	1949.0	39	695	4665	0.0	8.2	40
U-instances																					
$I10$	$Q5$	55	0	0.0	1.1	5	54	1	0.0	2.2	5	47	16	0.0	9.2	5	39	213	0.0	6.8	5
	$Q10$	55	0	0.0	1.2	5	54	1	0.0	1.2	5	42	28	0.0	4.0	5	30	410	0.0	0.5	5
	$Q5$	109	0	0.0	92.0	5	109	0	0.0	110.4	5	107	3	0.0	504.8	5	106	395	0.0	3.4	5
$I30$	$Q10$	71	0	0.0	6.4	5	70	1	0.0	8.7	5	63	17	0.0	25.4	5	55	384	0.0	15.8	5
	$Q30$	70	0	0.0	0.9	5	70	0	0.0	1.4	5	61	19	0.0	15.6	5	34	1243	0.0	1.4	5
	$Q5$	185	0	1.1	1659.6	3	185	0	1.1	1558.9	3	183	0	0.8	1786.5	3	183	590	0.0	10.3	5
$I50$	$Q10$	99	0	0.0	47.4	5	99	0	0.0	66.5	5	97	4	0.0	201.0	5	92	513	0.0	16.3	5
	$Q50$	75	0	0.0	1.8	5	75	0	0.0	3.6	5	68	21	0.0	12.8	5	35	1657	0.0	1.7	5
Cumulative		719	0	1.1	1810.5	38	716	3	1.1	1752.9	38	668	108	0.8	2559.2	38	574	5405	0.0	56.2	40

Table 7.4: Summary CPLEX results for formulation AS^c.

computing times and optimality gaps values for every size and capacity of D- and U- instances. The figures evidence of robust performance of formulation AS^C.

Furthermore, Figure 7.5, shows the computing time performance profile for both formulations. This figure presents, for each formulation and every type of instance, the percentage of optimal solutions obtained over the first 100 seconds of computing time. We can clearly observe that AS is outperformed by AS^c. In addition, we observe that the performance of the formulations not only depends on the type of instances but also on the β values. With AS, optimal solutions for D-instances are obtained in shorter times than those obtained for U-instances, specially for values of $\beta = 0.2$ and $\beta = 0.8$. This difference is more evident for AS^c in which the differences are observed for all values of β. We can thus conclude that AS^c, has the best performance over the formulations for the A-MSSP, independently of the type of instance.

7.4 Branch-and-price results

In order to analyze the performance of the BP algorithm proposed in Chapter 5.2, we present the numerical results of the computational experiments for the formulations MP and MP^c presented in Chapter 5.1. We use the benchmark instances as well as the computational environment described in Section 7.1. In particular, the BP was implemented within SCIP branch-and-price framework (Gamrath, 2010).

Table 7.5 gives a summary of the results obtained with SCIP for the BP algorithm with formulation MP. The table compares the results with those obtained with CPLEX for formulation AS. The results are presented for every value of β and every type of instances, D-

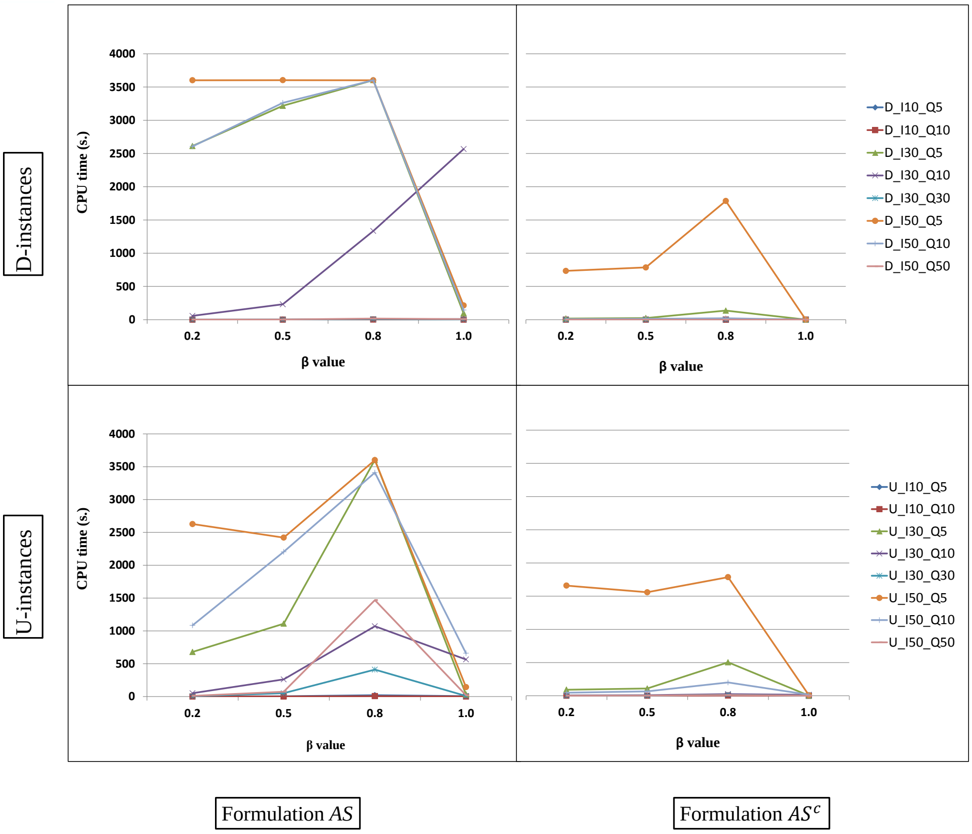


Figure 7.3: Average CPU times for the A-MSSP formulations.

and U-instances. Each table displays averages over the five instances, of the initial lower bounds (z_{LP}) (LP-relaxation for the AS and lower bound at the root node for MP) and the best solution found (UB). The columns labeled *Gap* show the average of the percentage relative deviations of the best-known solutions with respect to the lower bounds at termination (LB); that is, $Gap = \frac{UB-LB}{LB}100$. The average computing times in seconds are given in columns *Time*. The detailed results can be found in Tables C.1.1, C.1.2, C.1.3, C.1.4 in the Appendix C.1.

As can be seen, algorithm BP with MP is not able to provide initial lower bounds for most of the instances of $I50$. However, for smaller instances, we find some interesting results. Column generation provides better initial lower bounds than the LP-relaxation for formulation AS. The improvement is more evident for instances with smaller values of Q. In particular, the initial lower bounds for the MP are, on average, 25.77% and 40.81% higher than the LP-relaxation for formulation AS. BP with MP guarantees optimality for all solutions for U-instances. For D-instances, most of the upper bounds are also optimal, except for instances of $I30$ with $Q = 30$ and $\beta = \{0.5, 0.8\}$. For these instances, BP yields sizeable gaps.

Regarding computing times, BP with MP outperforms formulation AS in some cases. For D-instances, average computing times are smaller for instances of $I30$ and smaller values of Q. For U-instances, we observe smaller computing times for instances with $\beta = \{0.8, 1.0\}$. Observe that the improvement regarding computing times are specially attractive for instances with smaller values of Q. On the contrary, larger computing times can be observed for instances with larger values of Q. The above and the absence of lower and upper bounds for instances of $I50$ suggest the existence of symmetry problems, specifically, the symmetry for patterns within columns.

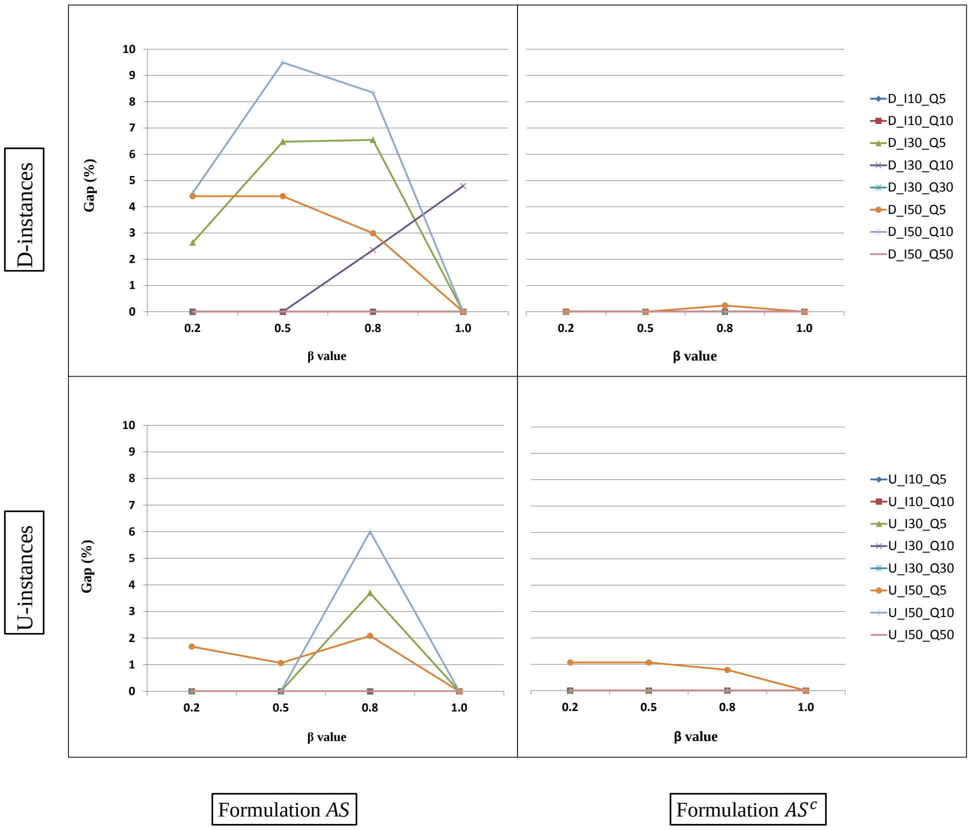


Figure 7.4: Average Gaps (%) for the A-MSSP formulations.

Table 7.6 gives a summary of the results obtained with SCIP for the BP algorithm with formulation MP^c. The table compares the results with those obtained with CPLEX for formulation AS^c. The results are presented in the same manner as for Table 7.5. The detailed results can be found in Tables C.2.1, C.2.2, C.2.3, C.2.4 in the Appendix C.2.

Contrary to BP algorithm with MP, BP with MP^c provides lower and upper bounds for all the instances, including instances of $I50$. In fact, upper bounds obtained with BP are optimal for all D-instances. For U-instances, optimal solutions are not guaranteed for most of the instances of $I50$ with $Q = 5$ and $\beta = \{0.2, 0.5, 0.8\}$. For these instances, BP with MP^c provides the same upper bounds as AS^c, with similar gaps and computing times. Regarding initial lower bounds, column generation for MP^c provides better initial lower bounds than the LP-relaxation for formulation AS^c. We can observe from Tables 7.5 and 7.6 that values for z_{LP} are almost the same for MP and MP^c. Different values can only be observed for D-instances of $I10$ with $Q = 10$ and $\beta = \{0.2, 0.5, 0.8\}$ (note also that, as can be seen in Tables 7.5 and 7.6, values for the LP-relaxations between formulations AS and AS^c are equal). In particular, the initial lower bounds for MP^c are, on average, 23.66% and 36.59% higher than the LP-relaxation of formulation AS^c. The difference between these percentages and the ones obtained with BP for MP is due to the little improvement of the initial lower bounds obtained with MP^c for instances of $I50$.

A very remarkable performance can be observed for D-instances of $I50$ with $Q = 5$ and $\beta = 0.8$. This set of instances are hard to solve for CPLEX with both formulations AS and AS^c. However, BP with MP^c provides optimal solutions with noticeable shorter computing times. For the remaining optimal solutions obtained with BP, smaller computing times are observed for D-instances with larger number of customers and smaller values of Q. For U-instances, smaller

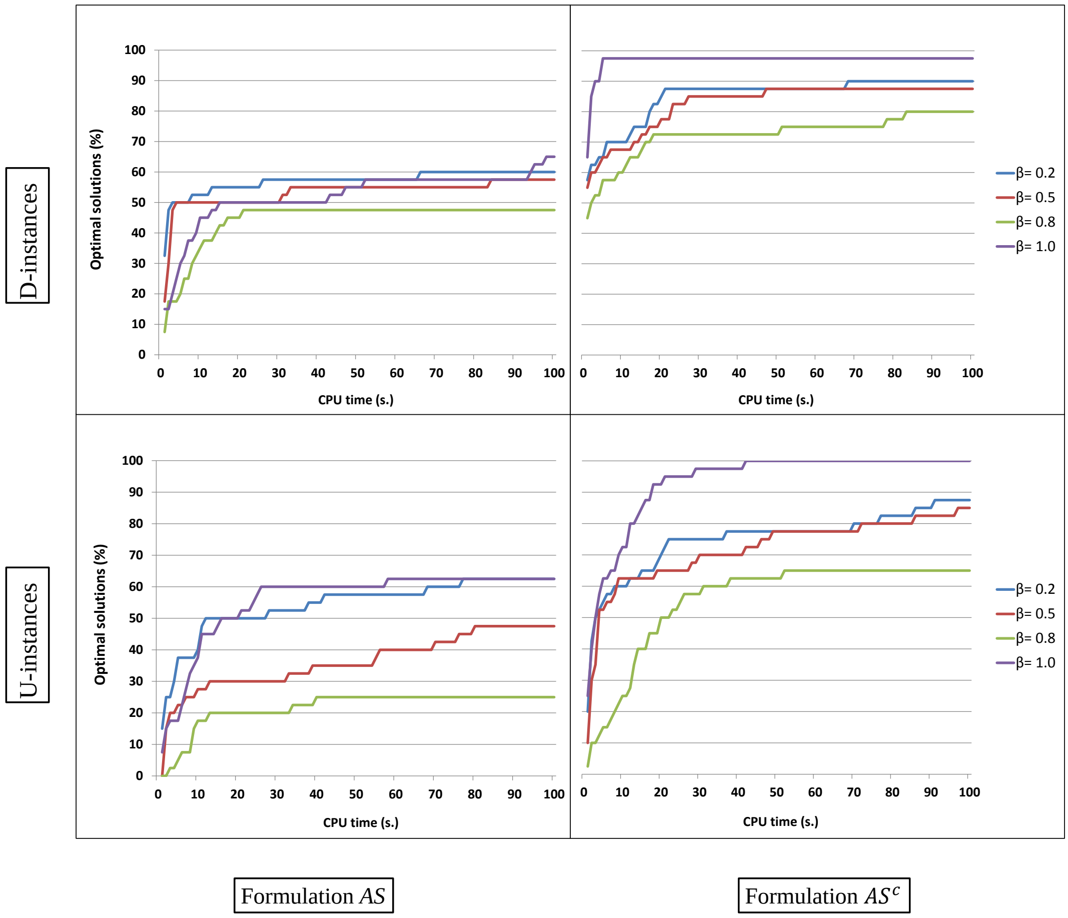


Figure 7.5: CPU time performance profile for the A-MSSP formulations.

computing times are observed only in some cases with values of $\beta = \{0.2, 1.0\}$.

From the above results we can observe that even if formulations MP and MP^c produce similar initial lower bounds, they show a noticeable difference on the convergence of the BP algorithm, specially for larger instances and smaller values of Q.

7.5 Computational results of the heuristic

To assess the effectiveness of the heuristic, we ran a series of computational experiments using the benchmark instances and the computational environment described in Section 7.1. For the local search procedure we set a number of 100 perturbed solutions. Even if we ran computational experiments using all five strategies described in Chapter 6.2.2, below we present the best strategy results for each policy. That is, strategies S_{AC} and S_{AB} for PS and AS policies, respectively.

Table 7.7 allows to perceive the contribution of each component of the heuristic algorithm to its overall performance for the PS and AS policies, over each group of D- and U-instances, for every value of I and Q. In particular, for each policy, we compare the results of the heuristic at the end of each phase with the values of the best solutions obtained with any of the formulations. Columns HG_G, HG_L, and HG_P denote the averages over the five instances in each group of the relative difference between the best-known solution obtained by CPLEX with any of the

β	Instance		D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP
0.2	$I10$	$Q5$	7.84	7.95	10.2	10.2	0.0	0.0	0.7	15.4	6.88	7.10	11.0	11.0	0.0	0.0	1.3	21.3
		$Q10$	3.92	5.97	8.0	8.0	0.0	0.0	0.3	13.7	3.44	6.28	11.0	11.0	0.0	0.0	0.6	9.9
	$I30$	$Q5$	25.56	25.56	28.2	28.0	2.6	0.0	2612.4	444.9	20.60	20.60	21.8	21.8	0.0	0.0	678.8	683.4
		$Q10$	12.78	12.80	15.8	15.8	0.0	0.0	57.5	556.6	10.30	10.34	14.2	14.2	0.0	0.0	50.2	807.5
		$Q30$	4.26	7.29	10.0	10.0	0.0	0.0	0.9	1536.5	3.43	7.31	14.0	14.0	0.0	0.0	4.0	372.0
	$I50$	$Q5$	46.48	-	48.6	-	4.4	-	3601.7	-	36.28	-	37.2	-	1.7	-	2627.7	-
		$Q10$	23.24	-	26.4	-	4.5	-	2605.8	-	18.14	-	19.8	-	0.0	-	1084.0	-
		$Q50$	4.65	-	10.0	-	0.0	-	1.6	-	3.63	-	15.0	-	0.0	-	10.3	-
0.5	$I10$	$Q5$	19.60	19.87	25.5	25.5	0.0	0.0	1.3	15.6	17.20	17.74	27.5	27.5	0.0	0.0	3.3	18.2
		$Q10$	9.80	14.93	20.0	20.0	0.0	0.0	0.4	14.6	8.60	15.69	27.5	27.5	0.0	0.0	1.5	10.2
	$I30$	$Q5$	63.90	63.90	70.0	70.0	6.5	0.0	3217.1	909.5	51.50	51.50	54.5	54.5	0.0	0.0	1109.1	1678.1
		$Q10$	31.95	32.00	39.5	39.5	0.0	0.0	232.0	1218.4	25.75	25.86	35.5	35.5	0.0	0.0	260.8	1368.4
		$Q30$	10.65	18.21	25.0	25.0	0.0	22.4	2.2	3408.4	8.58	17.77	35.0	35.0	0.0	0.0	50.8	928.5
	$I50$	$Q5$	116.20	-	121.5	-	4.4	-	3603.1	-	90.70	-	92.5	-	1.1	-	2420.5	-
		$Q10$	58.10	-	65.5	-	9.5	-	3261.9	-	45.35	-	49.5	-	0.0	-	2203.7	-
		$Q50$	11.62	-	25.0	-	0.0	-	2.7	-	9.07	-	37.5	-	0.0	-	72.8	-
0.8	$I10$	$Q5$	31.36	31.78	38.0	38.0	0.0	0.0	3.4	17.8	27.52	28.39	40.8	40.8	0.0	0.0	20.3	15.0
		$Q10$	15.68	23.89	31.2	31.2	0.0	0.0	1.3	25.7	13.76	25.11	39.2	39.2	0.0	0.0	6.0	9.2
	$I30$	$Q5$	102.24	102.24	109.4	109.4	6.5	0.0	3603.4	1306.0	82.40	82.40	86.2	86.2	3.7	0.0	3604.0	1713.5
		$Q10$	51.12	51.19	62.0	62.0	2.4	0.0	1333.8	1042.8	41.20	41.37	53.8	53.8	0.0	0.0	1070.4	986.4
		$Q30$	17.04	29.14	40.0	40.0	0.0	37.3	8.5	3600.0	13.73	28.43	52.6	52.6	0.0	0.0	410.1	600.7
	$I50$	$Q5$	185.92	-	191.6	-	3.0	-	3602.0	-	145.12	-	148.2	-	2.1	-	3601.7	-
		$Q10$	92.96	-	102.0	-	8.3	-	3602.6	-	72.56	-	78.8	-	6.0	-	3411.0	-
		$Q50$	18.59	-	40.0	-	0.0	-	16.7	-	14.51	-	58.6	-	0.0	-	1468.7	-
1.0	$I10$	$Q5$	39.20	40.40	43	43	0.0	0.0	10.3	7.5	34.40	36.16	39	39	0.0	0.0	5.5	7.6
		$Q10$	19.60	29.00	29	29	0.0	0.0	0.8	0.6	17.20	30.00	30	30	0.0	0.0	1.1	0.5
	$I30$	$Q5$	127.80	130.00	130	130	0.0	0.0	97.0	247.5	103.00	106.00	106	106	0.0	0.0	32.3	374.8
		$Q10$	63.90	64.70	70	70	4.8	0.0	2566.4	174.6	51.50	53.70	55	55	0.0	0.0	564.6	264.6
		$Q30$	21.30	35.00	35	35	0.0	0.0	4.7	1.1	17.17	34.00	34	34	0.0	0.0	8.1	1.2
	$I50$	$Q5$	232.40	-	234	-	0.0	-	214.8	-	181.40	-	183	-	0.0	-	142.9	-
		$Q10$	116.20	-	119	-	0.0	-	144.3	-	90.70	-	92	-	0.0	-	659.5	-
		$Q50$	23.24	35.00	35	35	0.0	0.0	8.9	1.7	18.14	35.00	35	35	0.0	0.0	19.8	1.9

Table 7.5: Branch-and-price results comparison for formulation MP.

formulations and the value of the solution found by the greedy procedure, the local search applied to the greedy solution, and the best value after applying the local search, respectively, over 100 perturbed solutions. The detailed results can be found in Tables D.1.1 and D.1.2 in the Appendix D.1. As can be seen, the local search produces a significant improvement over the greedy solution, particularly for $\beta = 0.8$ when the greedy solution usually provides poor quality solutions. On the other hand, we can observe that the repetition over the perturbed solutions is significant for closing the gaps.

For each $|I|$ and Q, Table 7.8 shows the summary of the results for the PS and AS policies for the D- and U-instances. Columns z and e display the total number of operators used and the total earliness, respectively, over the five instances. We use columns *Time* to display the average computing times needed by the heuristic in seconds. The detailed results can be found in Tables D.2.1 and D.2.2 in the Appendix D.2.

Next, we compare the results of the heuristic with those obtained by CPLEX for the MILP formulations presented in Chapters 3 and 4, for the P-MSSP and A-MSSP, respectively. For the PS policy, a summary comparison for the P-MSSP formulations is given in Table 7.9. Columns *HGap* denote the relative difference (in %) between the best solution found by CPLEX for the corresponding formulation and the value of the best solution found by the heuristic, averaged over the five instances. Negative values indicate that the heuristic yields better solutions than the time constrained CPLEX. We use columns *HTime* to display the difference (in seconds) between the time required by the heuristic and the time required by CPLEX, averaged over the

β	Instance		D-instances z_{LP} AS^c	MP^c	UB AS^c	MP^c	Gap AS^c	MP^c	$Time$ AS^c	MP^c	U-instances z_{LP} AS^c	MP^c	UB AS^c	MP^c	Gap AS^c	MP^c	$Time$ AS^c	MP^c
0.2	$I10$	$Q5$	7.84	7.95	10.2	10.2	0.0	0.0	0.4	1.4	6.88	7.10	11.0	11.0	0.0	0.0	1.1	17.8
		$Q10$	3.92	6.37	8.0	8.0	0.0	0.0	0.2	0.7	3.44	6.28	11.0	11.0	0.0	0.0	1.2	17.3
	$I30$	$Q5$	25.56	25.56	28.0	28.0	0.0	0.0	17.2	67.5	20.60	20.60	21.8	21.8	0.0	0.0	92.0	197.9
		$Q10$	12.78	12.80	15.8	15.8	0.0	0.0	1.5	6.7	10.30	10.34	14.2	14.2	0.0	0.0	6.4	34.6
		$Q30$	4.26	7.29	10.0	10.0	0.0	0.0	0.2	1.0	3.43	7.11	14.0	14.0	0.0	0.0	0.9	55.0
	$I50$	$Q5$	46.48	46.48	48.4	48.4	0.0	0.0	733.9	21.8	36.28	36.28	37.0	37.0	1.1	1.1	1659.6	2119.8
		$Q10$	23.24	23.24	26.4	26.4	0.0	0.0	8.2	64.5	18.14	18.14	19.8	19.8	0.0	0.0	47.4	220.9
		$Q50$	4.65	7.29	10.0	10.0	0.0	0.0	0.2	1.1	3.63	7.40	15.0	15.0	0.0	0.0	1.8	61.7
0.5	$I10$	$Q5$	19.60	19.87	25.5	25.5	0.0	0.0	0.3	1.6	17.20	17.74	27.5	27.5	0.0	0.0	2.2	60.4
		$Q10$	9.80	15.93	20.0	20.0	0.0	0.0	0.3	1.5	8.60	15.69	27.5	27.5	0.0	0.0	1.2	121.2
	$I30$	$Q5$	63.90	63.90	70.0	70.0	0.0	0.0	24.8	84.5	51.50	51.50	54.5	54.5	0.0	0.0	110.4	184.5
		$Q10$	31.95	32.00	39.5	39.5	0.0	0.0	1.6	11.2	25.75	25.86	35.5	35.5	0.0	0.0	8.7	65.1
		$Q30$	10.65	18.21	25.0	25.0	0.0	0.0	0.2	2.0	8.58	17.77	35.0	35.0	0.0	0.0	1.4	288.3
	$I50$	$Q5$	116.20	116.20	121.0	121.0	0.0	0.0	786.2	21.8	90.70	90.70	92.5	92.5	1.1	1.4	1558.9	2376.7
		$Q10$	58.10	64.46	65.5	65.5	0.0	0.0	13.3	83.8	45.35	45.35	49.5	49.5	0.0	0.0	66.5	253.4
		$Q50$	11.62	18.21	25.0	25.0	0.0	0.0	0.2	1.7	9.07	18.50	37.5	37.5	0.0	0.0	3.6	139.5
0.8	$I10$	$Q5$	31.36	31.78	38.0	38.0	0.0	0.0	0.8	3.1	27.52	28.39	40.8	40.8	0.0	0.0	2.2	57.0
		$Q10$	15.68	25.29	31.2	31.2	0.0	0.0	0.6	6.2	13.76	25.11	39.2	39.2	0.0	0.0	1.2	159.5
	$I30$	$Q5$	102.24	102.24	109.4	109.4	0.0	0.0	136.3	88.5	82.40	82.40	86.2	86.2	0.0	0.0	110.4	1367.1
		$Q10$	51.12	51.19	62.0	62.0	0.0	0.0	4.2	35.2	41.20	41.37	53.8	53.8	0.0	0.0	8.7	217.0
		$Q30$	17.04	29.14	40.0	40.0	0.0	0.0	0.2	6.7	13.73	28.43	52.6	52.6	0.0	0.0	1.4	1010.2
	$I50$	$Q5$	185.92	185.92	191.2	191.2	1.2	0.0	1785.5	27.5	145.12	145.12	146.4	146.4	1.1	0.8	1558.9	2062.0
		$Q10$	92.96	92.96	102.0	102.0	0.0	0.0	21.1	130.8	72.56	72.56	78.4	78.4	0.0	0.0	66.5	726.2
		$Q50$	18.59	29.14	40.0	40.0	0.0	0.0	0.4	3.1	14.51	29.60	58.6	58.6	0.0	0.0	3.6	906.5
1.0	$I10$	$Q5$	39.20	40.40	43.0	43.0	0.0	0.0	0.8	1.2	34.40	36.16	39.0	39.0	0.0	0.0	6.8	9.4
		$Q10$	19.60	29.00	29.0	29.0	0.0	0.0	0.2	0.5	17.20	30.00	30.0	30.0	0.0	0.0	0.5	1.4
	$I30$	$Q5$	127.80	130.00	130.0	130.0	0.0	0.0	0.8	1.0	103.00	106.00	106.0	106.0	0.0	0.0	3.4	2.9
		$Q10$	63.90	64.70	70.0	70.0	0.0	0.0	3.5	13.1	51.50	53.70	55.0	55.0	0.0	0.0	15.8	137.9
		$Q30$	21.30	35.00	35.0	35.0	0.0	0.0	0.2	1.0	17.17	34.00	34.0	34.0	0.0	0.0	1.4	2.8
	$I50$	$Q5$	232.40	234.00	234.0	234.0	0.0	0.0	1.4	1.3	181.40	183.00	183.0	183.0	0.0	0.0	10.3	3.9
		$Q10$	116.20	119.00	119.0	119.0	0.0	0.0	0.9	1.1	90.70	92.00	92.0	92.0	0.0	0.0	16.3	3.4
		$Q50$	23.24	35.00	35.0	35.0	0.0	0.0	0.2	0.6	18.14	35.00	35.0	35.0	0.0	0.0	1.7	2.5

Table 7.6: Branch-and-price results comparison for formulation MP^c.

five instances. The detailed values can be found in Tables D.3.1 and D.3.2 in the Appendix D.3.

Relative to sparse formulations, we can make the following observations: the heuristic obtains optimal solutions for all instances in which CPLEX does for formulations PS and PS^c. For instances not optimally solved by CPLEX, the heuristic produces the same (or a better) upper bound with significantly less time, specially for larger instances and small values of Q. Relative to dense formulations, the heuristic is outperformed by both formulations PD and PD^c. Moreover, the heuristic is not able to find optimal solutions for some of the harder instances for which CPLEX does (instances of "$I50$" with $Q = 5$). In addition, computing times for the optimal solutions found by the heuristic are larger than those of the dense formulations.

A summary comparison of the heuristic and the A-MSSP formulations is given in Table 7.10. The table displays the results for each value of β and for each series of instances. The meaning of the columns in the tables is the same as in Table 7.9. The detailed values can be found in Tables D.4.1 and D.4.2 in the Appendix D.4.

The heuristic solutions are, on average, 0.16% and 0.38% worse than CPLEX, with a maximum relative deviation of 5.00% and 5.36% for the D- and U-instances, respectively. As for the different values of β, for $\beta = 1.0$ the heuristic finds the same solutions as CPLEX, 100% of which were proven to be optimal with formulation AS^c. The same results are obtained for $\beta = 0.2$ and $\beta = 0.5$, 100% and 92.5% of which were proven to be optimal, for the D- and U-instances, respectively. The worst outcome of the heuristic is observed for $\beta = 0.8$, with average *HGaps* of 0.75% and 1.52% for the D- and U-instances, respectively. However, in spite of this,

		PS policy			AS policy											
					$\beta=0.2$			$\beta=0.5$			$\beta=0.8$			$\beta=1.0$		
		HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P
D-instances																
$I10$	$Q5$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.1	1.0	0.0	5.0	2.5	0.0
	$Q10$	6.0	0.0	0.0	2.0	0.0	0.0	6.0	0.0	0.0	6.7	0.0	0.0	0.0	0.0	0.0
$I30$	$Q5$	3.4	0.7	0.0	2.7	0.7	0.0	3.4	0.0	0.0	14.3	1.6	1.6	4.6	3.9	0.0
	$Q10$	9.0	2.6	0.0	6.4	0.0	0.0	9.0	1.2	0.0	37.7	2.3	2.3	5.8	1.4	0.0
	$Q30$	10.0	0.0	0.0	6.0	0.0	0.0	10.0	0.0	0.0	48.5	12.0	0.0	0.0	0.0	0.0
$I50$	$Q5$	2.5	0.8	0.0	1.7	0.5	0.0	2.5	0.5	0.0	5.6	0.3	0.1	2.1	2.1	0.0
	$Q10$	6.0	2.2	0.0	3.8	0.0	0.0	6.8	2.2	0.0	39.3	2.1	2.1	5.9	5.9	0.0
	$Q50$	10.0	0.0	0.0	2.0	0.0	0.0	10.0	0.0	0.0	71.5	29.5	0.0	0.0	0.0	0.0
U-instances																
$I10$	$Q5$	7.6	0.0	0.0	5.9	0.0	0.0	7.6	0.0	0.0	18.1	0.0	0.0	2.5	2.5	0.0
	$Q10$	7.6	0.0	0.0	5.9	0.0	0.0	7.6	0.0	0.0	12.2	0.0	0.0	0.0	0.0	0.0
$I30$	$Q5$	7.6	2.9	0.0	9.0	2.9	0.0	8.4	4.6	0.0	38.8	3.1	3.1	3.9	3.0	0.0
	$Q10$	10.1	1.7	0.0	8.4	1.7	0.0	10.1	4.4	0.0	62.9	7.7	2.5	14.9	5.3	0.0
	$Q30$	5.9	0.0	0.0	5.9	0.0	0.0	5.9	0.0	0.0	37.6	10.2	2.1	0.0	0.0	0.0
$I50$	$Q5$	7.6	3.2	1.1	8.2	2.1	0.0	8.1	5.4	0.0	33.9	2.4	0.7	2.8	2.8	0.0
	$Q10$	13.2	6.1	0.0	11.2	5.2	0.0	13.2	10.1	0.0	109.4	4.3	2.3	8.8	5.5	0.0
	$Q50$	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	83.2	11.2	2.4	0.0	0.0	0.0

Table 7.7: Average relative contribution of each heuristic phase.

		PS Policy		AS policy											
				$\beta=0.2$			$\beta=0.5$			$\beta=0.8$			$\beta=1.0$		
		z	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$
D-instances															
$I10$	$Q5$	51	1.9	51	0	3.0	51	0	3.0	46	6	2.5	43	69	2.1
	$Q10$	40	2.9	40	0	2.9	40	0	3.0	37	8	4.8	29	200	2.2
$I30$	$Q5$	140	9.1	140	0	28.7	140	0	66.4	137	8	221.5	130	270	27.5
	$Q10$	79	15.7	79	0	20.7	79	0	40.2	78	5	43.7	70	337	30.1
	$Q30$	50	13.7	50	0	24.3	50	0	29.1	50	0	36.5	35	999	28.4
$I50$	$Q5$	242	30.3	242	0	28.3	242	0	53.2	238	5	58.2	234	453	45.1
	$Q10$	132	56.4	132	0	57.0	131	0	56.6	128	9	89.1	119	485	44.5
	$Q50$	50	58.7	50	0	23.7	50	0	72.9	50	0	65.9	35	1356	62.7
U-instances															
$I10$	$Q5$	55	1.8	55	0	3.2	55	0	3.5	49	8	3.0	39	139	2.1
	$Q10$	55	3.8	55	0	3.6	55	0	3.9	47	8	5.8	30	169	2.4
$I30$	$Q5$	109	9.5	109	0	47.7	109	0	58.3	109	8	58.7	106	322	36.1
	$Q10$	71	42.5	71	0	20.2	71	0	66.5	67	8	33.9	55	384	29.8
	$Q30$	70	26.8	70	0	23.5	70	0	56.8	65	9	42.8	34	771	49.0
$I50$	$Q5$	185	48.7	185	0	108.6	185	0	82.8	184	1	9.4	183	529	51.6
	$Q10$	99	75.7	99	0	68.4	99	0	65.7	99	5	152.1	92	490	62.1
	$Q50$	75	56.3	75	0	41.6	75	0	219.4	72	12	219.2	35	1054	130.5

Table 7.8: Summary of heuristic results for the D- and U-instances.

for larger instances ("$I30$" and "$I50$") and no trivial capacities ($Q \neq I$), the computing times of the heuristic are in some cases noticeably shorter than those of CPLEX. The same behaviour can also be observed for values of $\beta \in \{0.2, 0.5\}$.

In summary, the heuristic produces good quality solutions in short cpu times, for the A-MSSP, particularly for larger instances and non-trivial capacities, which are the most difficult ones for CPLEX. Nevertheless, the heuristic is not competitive relative to the P-MSSP formulations, which is the most effective formulations for the periodic case. The explanation can be found in the high effectiveness of formulations PD and PDc, which obtain optimal solutions in very short amounts of computing time. Additionally, the heuristic needs to apply a local search procedure to perturbed solutions in order to close the gaps, which reduces its efficiency.

		Sparse formulations				Dense formulations			
		PS		PS^c		PD		PD^c	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
D-instances									
$I10$	$Q5$	0.0	1.1	0.0	1.7	0.0	1.8	0.0	1.9
	$Q10$	0.0	2.8	0.0	2.8	0.0	2.7	0.0	2.8
	$Q5$	0.0	-1558.5	0.0	-2.0	0.0	6.0	0.0	7.7
$I30$	$Q10$	0.0	-3.5	0.0	15.1	0.0	15.5	0.0	15.6
	$Q30$	0.0	13.6	0.0	13.6	0.0	13.6	0.0	13.6
	$Q5$	-0.5	-3571.7	0.0	-160.6	0.0	-336.9	0.0	15.3
$I50$	$Q10$	0.0	-1327.5	0.0	51.6	0.0	54.4	0.0	55.7
	$Q50$	0.0	58.6	0.0	58.6	0.0	58.6	0.0	58.6
U-instances									
$I10$	$Q5$	0.0	1.3	0.0	1.2	0.0	1.7	0.0	1.7
	$Q10$	0.0	3.6	0.0	3.6	0.0	3.7	0.0	3.6
	$Q5$	0.0	-314.8	0.0	-31.5	0.0	7.2	0.0	7.4
$I30$	$Q10$	0.0	31.2	0.0	40.7	0.0	42.2	0.0	42.2
	$Q30$	0.0	26.6	0.0	26.6	0.0	26.5	0.0	26.6
	$Q5$	-0.5	-2306.5	0.0	-1423.1	1.1	23.9	1.1	-34.6
$I50$	$Q10$	0.0	-287.1	0.0	48.7	0.0	73.1	0.0	74.4
	$Q50$	0.0	55.9	0.0	56.1	0.0	55.8	0.0	56.1

Table 7.9: Summary heuristic comparison for the P-MSSP formulations.

		AS								AS^c							
		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
D-instances																	
$I10$	$Q5$	0.0	2.3	0.0	0.6	0.0	-0.0	0.0	-0.0	0.0	2.6	0.0	0.9	0.0	0.7	0.0	0.7
	$Q10$	0.0	2.6	0.0	2.6	0.0	3.5	0.0	1.4	0.0	2.7	0.0	2.7	0.0	4.3	0.0	2.0
	$Q5$	-0.6	-2583.7	0.0	-3150.7	1.6	-3381.9	0.0	-69.5	0.0	11.5	0.0	41.6	1.6	85.3	0.0	26.7
$I30$	$Q10$	0.0	-36.7	0.0	-191.8	2.3	-1290.1	0.0	-2536.2	0.0	19.2	0.0	38.6	2.3	39.5	0.0	26.6
	$Q30$	0.0	23.4	0.0	26.9	0.0	28.0	0.0	23.7	0.0	24.1	0.0	28.8	0.0	36.3	0.0	28.2
	$Q5$	-0.4	-3573.5	-0.4	-3549.9	-0.1	-3543.9	0.0	-169.7	0.0	-705.6	0.0	-733.0	0.1	-1727.3	0.0	43.7
$I50$	$Q10$	0.0	-2548.8	0.0	-3205.4	2.1	-3513.6	0.0	-99.8	0.0	48.8	0.0	43.3	2.1	68.0	0.0	43.5
	$Q50$	0.0	22.1	0.0	70.3	0.0	49.2	0.0	53.8	0.0	23.5	0.0	72.7	0.0	65.5	0.0	62.5
U-instances																	
$I10$	$Q5$	0.0	2.0	0.0	0.1	0.0	-0.8	0.0	-0.3	0.0	2.1	0.0	0.4	0.0	-0.4	0.0	-0.2
	$Q10$	0.0	2.9	0.0	2.3	0.0	-0.2	0.0	1.4	0.0	2.4	0.0	2.7	0.0	1.9	0.0	2.0
	$Q5$	0.0	-631.2	0.0	-1050.9	3.1	-3545.3	0.0	3.8	0.0	-44.4	0.0	-52.1	3.1	-446.1	0.0	32.7
$I30$	$Q10$	0.0	-30.0	0.0	-194.3	2.5	-1036.5	0.0	-534.8	0.0	13.8	0.0	57.8	2.5	8.5	0.0	14.0
	$Q30$	0.0	19.4	0.0	5.9	2.1	-367.3	0.0	40.9	0.0	22.5	0.0	55.3	2.1	27.2	0.0	47.6
	$Q5$	-0.5	-2519.1	0.0	-2337.8	-0.5	-3592.3	0.0	-91.4	0.0	-1551.1	0.0	-1476.1	0.7	-1777.1	0.0	41.2
$I50$	$Q10$	0.0	-1015.6	0.0	-2138.0	1.8	-3258.8	0.0	-597.4	0.0	21.0	0.0	-0.8	2.3	-48.8	0.0	45.8
	$Q50$	0.0	31.3	0.0	146.6	2.4	-1249.5	0.0	110.7	0.0	39.9	0.0	215.7	2.4	206.4	0.0	128.8

Table 7.10: Summary heuristic comparison for the A-MSSP formulations.

Chapter 8

Comparison of the policies for the MSSP

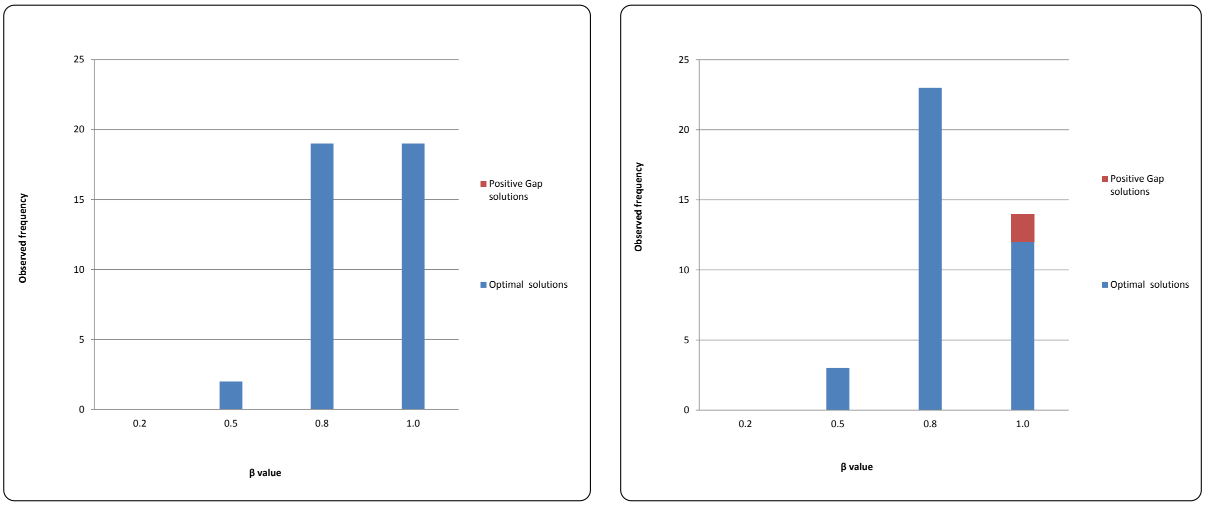
The last chapter of this thesis focuses on the implications derived from the modeling hypotheses of the two studied scheduling policies. In particular, this chapter is devoted to analyze the structure of the solutions obtained with each of the two studied policies and to evaluate the trade-off between them. The chapter is divided in two sections. In the first one, we present an analysis on the impact of the parameter β for the A-MSSP. In the second section, we compare indicators associated with solutions for both scheduling policies in order to evaluate the suitability of each of them.

8.1 Impact of the parameter β on the A-MSSP

As observed in Chapter 7.3, the parameter β has a crucial impact on the solutions of the formulations for the A-MSSP. To better perceive the effect of this parameter, Figure 8.1 displays the frequency chart that shows the smallest value of β for which the solution to an instance does apply earliness. The results presented in this chart correspond to optimal or best-known solutions for the A-MSSP. The frequencies are split into two subsets, proven optimal solutions and solutions with a positive *Gap*. We observe that for $\beta = 0.2$ earliness is not applied at all in any of the best solutions found. Ahead of time visits first appear in solutions for $\beta = 0.5$. Therefore, the computational experience suggests that earliness is only worth if the cost per operator is at least the same as the penalty for early visits. We observe that for almost 50% of the D-instances, earliness is only presented for $\beta = 1.0$, i.e., only when early visits are not penalized in the objective function. However, this percentage is only 35% for the U-instances. In particular, the average values for this "earliness breakpoint" are $\bar{\beta}_{bp} = 0.88$ and $\bar{\beta}_{bp} = 0.85$ for the D- and U-instances, respectively. This seems to suggest that the higher number of different service intervals for the customers, the less visits ahead of time appear in the solutions to the A-MSSP.

8.2 Comparison of solutions with PS and AS policies

In Figure 8.2 we present a comparative chart of the number of operators used over all periods with the PS policy and the AS policy, for the different values of β. For this comparison we have considered the sets of instances where the best-known solutions were proven optimal for every value of β. For values of $\beta \leq 0.5$ we observe the same number of operators for solutions with both service policies. This implies, together with Figure 8.1, that for values of $\beta \leq 0.5$, the P-MSSP and the A-MSSP have the same solutions, given that that we obtain the same number

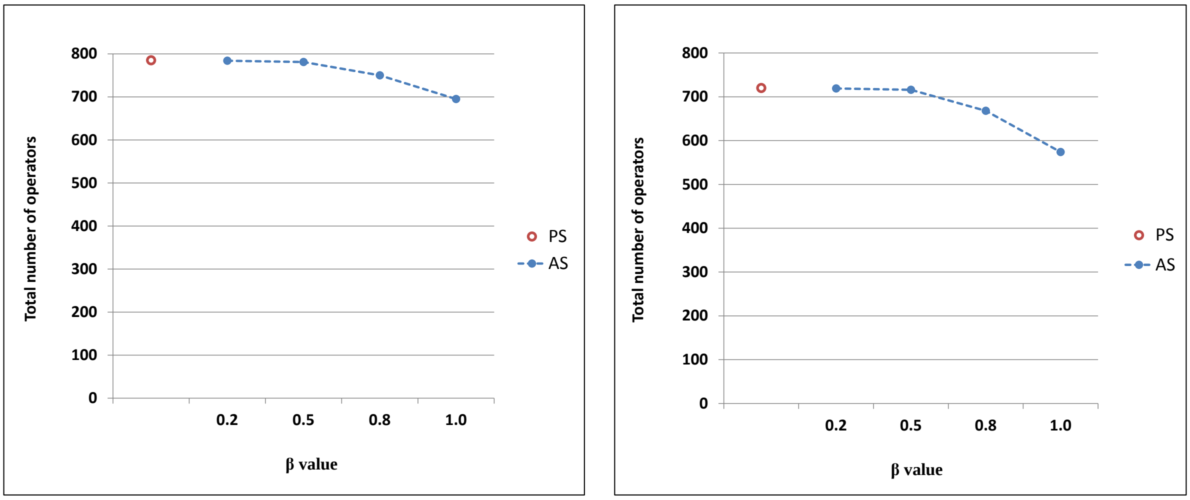


a) D-instances

b) U-instances

Figure 8.1: Observed earliness breakpoint for the A-MSSP solutions.

of operators and there are no early visits in these solutions. We can observe, however, that the number of operators used with the AS policy decreases clearly for values of $\beta > 0.5$.



a) D-instances

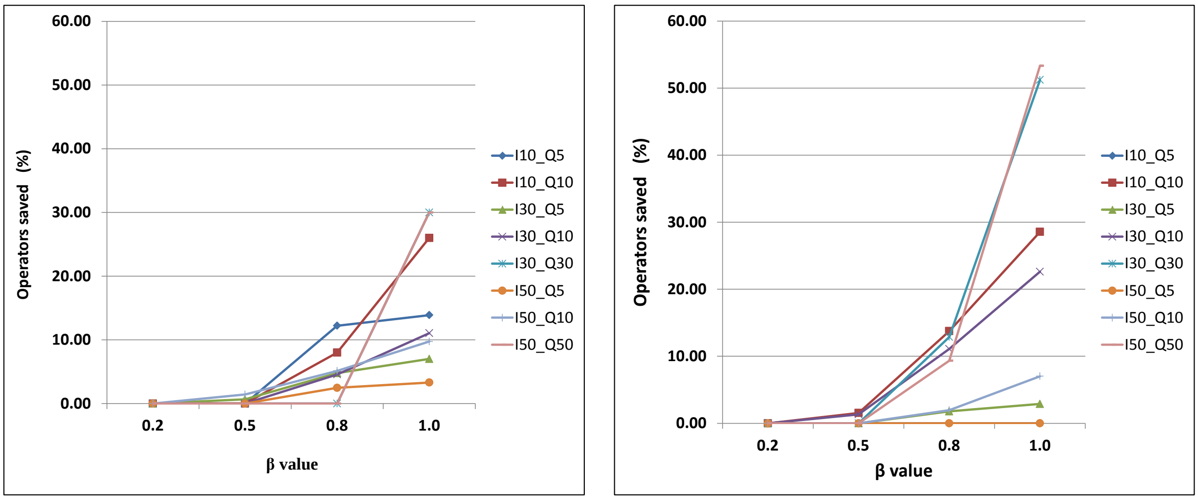
b) U-instances

Figure 8.2: Total number of operators for optimally solved instances.

We next analyze the savings on the number of operators that can be obtained by applying the AS policy. Table 8.1 displays the percentage of operators saved with respect to the PS policy. The entries of the table are average percentage savings over all the instances of the same type and for the different values of β. We observe that the savings noticeably increase as the value of β does. In particular, for $\beta = 0.2$, the AS policy does not produce any saving, however, the highest values are for $\beta = 1.0$, which presents, on average, 16.38% and 26.48% of the operators saved, for the D- and U-instances, respectively. In addition, the type of instance has an important impact on this percentages. We can observe that the number of operators that can be saved for the U-instances are twice the values for the D-instances. Moreover, we observe higher percentages for instances with larger values of I and larger values of Q. On the contrary, the lower percentages are for instances with larger values of I and smaller values of Q. The savings behaviour can be visually appreciated in Figure 8.3.

Instance		D-instances				U-instances			
		β value							
		0.2	0.5	0.8	1.0	0.2	0.5	0.8	1.0
$I10$	$Q5$	0.00	0.00	12.22	13.89	0.00	1.54	13.75	28.56
	$Q10$	0.00	0.00	8.00	26.00	0.00	1.54	23.36	46.22
	$Q5$	0.00	0.67	4.78	7.01	0.00	0.00	1.78	2.89
$I30$	$Q10$	0.00	0.00	4.58	11.05	0.00	1.33	11.10	22.62
	$Q30$	0.00	0.00	0.00	30.00	0.00	0.00	12.87	51.23
	$Q5$	0.00	0.00	2.47	3.31	0.00	0.00	0.00	0.00
$I50$	$Q10$	0.00	1.43	5.14	9.74	0.00	0.00	1.95	7.02
	$Q50$	0.00	0.00	0.00	30.00	0.00	0.00	9.33	53.33
Average		0.00	0.26	4.65	16.38	0.00	0.55	9.27	26.48

Table 8.1: Operators saved (in %) with AS policy.



a) D-instances b) U-instances

Figure 8.3: Operators saved (in %) with AS policy.

Figure 8.4 displays the average computing times for finding the best solutions for both the PS and AS policies. We can observe that the P-MSSP is actually easier to solve than the A-MSSP, for almost all values of β. Regarding the AS policy, we observe that the difficulty of the A-MSSP increases with the value of β. The only exception is $\beta = 1.0$. In this case, where we do not penalize earliness, all instances were solved to optimality with the smallest computational effort, even if in the comparison we also include the P-MSSP. On the other hand, we observe that, despite the fact that the A-MSSP solutions coincide with to those of P-MSSP for values of $\beta \leq 0.5$, the computational effort needed to obtain solutions to the A-MSSP is substantially higher than the computing time needed to obtain solutions to the P-MSSP. In other words, the use of the AS policy is only relevant for the cases in which the cost of employing a operator is cheaper than the cost associated to visit customers one period earlier.

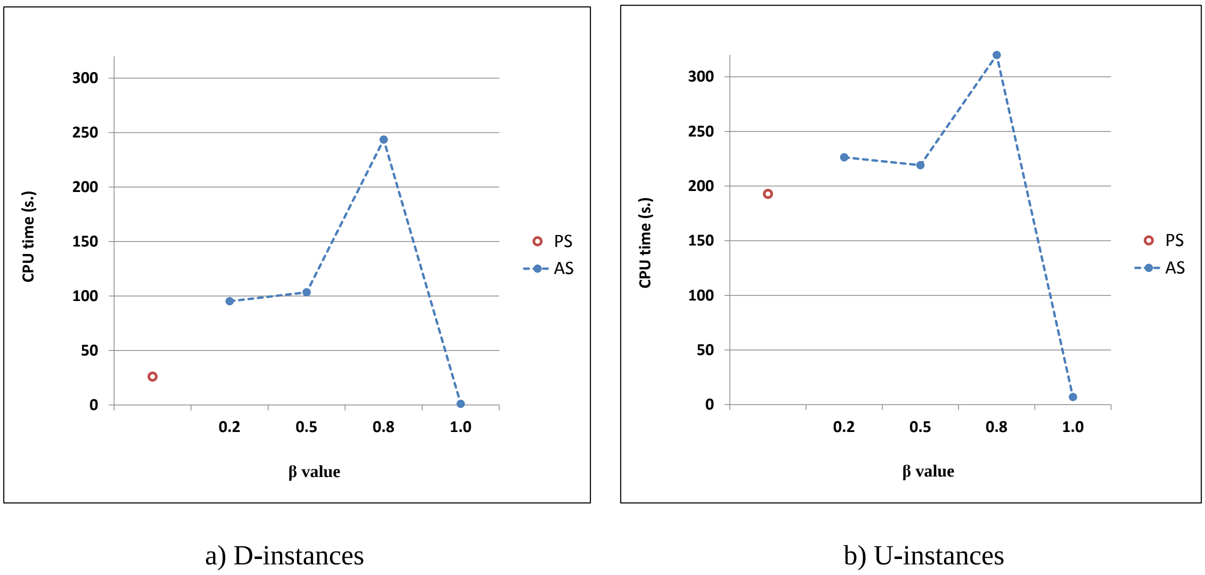


Figure 8.4: Average CPU times for the best-found solutions.

Chapter 9

Conclusions and Future Research

Decisions involving recurrent services can be found in a large variety of processes. The planning of such activities requires to determine the periods in which these services have to take place, that is, the scheduling. In this thesis, we have studied MSSPs, a family of problems in which the time duration for the services does not exceed one single period. The goal in the MSSPs is to reduce the total number of operators used over a planning horizon.

For these problems, we focus on two alternative scheduling policies. In the P-MSSP, where we apply the PS policy, the time intervals between two consecutive service periods of a customer are always of the same length and coincide with its service interval. In the A-MSSP, where we apply the AS policy, the time intervals between two consecutive service periods of a customer are not established in advance and can be of different lengths. In this policy customers may be visited ahead of time, however, is not allowed that the time between two consecutive visits to the same customer exceeds his service interval.

For the P-MSSP, we have developed four alternative MILP formulations: two sparse and two dense. For each of them we have developed a customer-based and a class-based formulation. The computational experience shows a noticeable better performance of the dense formulations. In particular, CPLEX guaranteed optimal solutions for all the tested instances. Additionally, class-based formulations have outperformed customer-based formulations, specially for the D-instances.

For the A-MSSP we have developed two alternative sparse MILPs, one customer-based and one class-based. In these formulations we have resorted to the earliness concept, that is the total number of early periods within the visits of the customers. To reduce the negative effect that such early services may cause, we minimize a weighted sum of the total number of operators and the total earliness. We have analyzed both formulations, deriving some reinforcements. The numerical results show that the class-based formulation provides better results that the customer-based formulation, particularly for the D-instances.

Because, A-MSSP formulations turned out to be harder to solve than MILPs for P-MSSP, we have proposed two alternative MILPs, which are CG reformulations of the sparse A-MSSP. For both formulations we have provided their corresponding pricing problem to generate new columns. We have also developed an initialization procedure in order to provide good quality initial solutions. Both CG formulations have been embedded within a BP framework, in which we combined several branching strategies. We have also included a procedure to recover infeasibility after branching. Furthermore, we tried several stabilization techniques in order forbid the generation of too many columns and therefore to speed up the algorithm.

The computational experience shows that CG formulations give better initial lower bounds than the LP-relaxation of the sparse formulations. A very interesting fact is that both formulations MP and MP^c provide almost the same initial lower bounds, for the cases in which MP is able to provide them. In spite of this, both CG formulations show a noticeable difference on the convergence of the BP algorithm, specially for larger instances and smaller values of Q. In particular, the numerical results of BP with MP formulation are not very encouraging. For most of the cases the computing times with MP are not competitive as compared to the sparse MILPs. In addition, for some of the larger instances and larger capacities, the program terminated because of the CPU time limit without providing a initial lower bound by solving the root node.

On the contrary, BP with MP^c provides lower and upper bounds for all the instances, including the largest ones. Moreover, the upper bounds obtained with MP^c are optimal for all D-instances. For U-instances, optimal solutions are not guaranteed for some of the larger instances and small capacities. However, for these instances, BP with MP^c provides the same upper bounds as its corresponding sparse MILP, producing similar gaps and computing times.

The effectiveness of the BP with MP^c relies on the remarkable performance that is observed for D-instances in which the A-MSSP is hard to solve with both sparse formulations, specially for larger instances and small capacities. For these instances BP with MP^c provides optimal solutions with considerably shorter computing times. This suggests that BP with MP may be preferred to sparse formulations for even larger instances and smaller capacities, which are in fact the most difficult instances to solve with sparse MILP formulations.

As an alternative solution technique for the MSSP we have also proposed a heuristic algorithm suitable for both of the scheduling policies. The algorithm is divided into a greedy and a local search procedure. For the LS we have tried several search strategies using two different neighborhoods. Additionally, we apply a diversification procedure which is crucial for closing the gaps. The computational experience shows that, for the PS policy, the heuristic results are outperformed by formulation PD^c, which produces optimal solutions within shorter CPU times. For the AS policy, the heuristic results are very competitive for larger instances and smaller capacities.

The solutions for the two alternative scheduling policies have been analyzed, concluding that the AS policy is only profitable in cases for which the cost of a single operator is cheaper than the cost of associated with visiting a customer one period before the end of his service interval. The numerical results show that using the AS policy may reduce the number of operators up to 26.48%, depending on the type of instance and the value of β.

This thesis has produced the following research papers:

- Scheduling policies for multi-period services. Cristina Núñez, Elena Fernández, Jörg Kalcsics, Stefan Nickel. *European Journal of Operational Research.* (Accepted).

- A branch-and-price for the Multi-period Service Scheduling Problem. Cristina Núñez, Elena Fernández, Jörg Kalcsics. *In preparation.*

- Improved formulations for the Multi-period Service Scheduling Problem. Cristina Núñez, Elena Fernández, Jörg Kalcsics. *In preparation.*

In addition, the results of this thesis have appeared in several presentations of international conferences and workshops as enlisted below:

- *Scheduling policies for periodic collection with balancing constraints.* Cristina Núñez, Elena Fernández, Stefan Nickel, Jörg Kalcsics. 4th Workshop on Combinatorial Optimization, Routing and Location (CORAL). Benicassim, Spain, May 2012.

- *Scheduling policies for periodic collection.* Cristina Núñez, Fernández, Stefan Nickel, Jörg Kalcsics. 1st Meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (Verolog). Bologna, Italy, June 2012.

- *Policies for the Multi-period Collection Scheduling Problem.* Cristina Núñez, Elena Fernández, Stefan Nickel, Jörg Kalcsics. 4th International Workshop on Locational Analysis and Related Problems (RedLoca). Torremolinos, Spain, June 2013.

- *The Multi-period Collection Scheduling Problem.* Elena Fernández, Cristina Núñez, Stefan Nickel, Jörg Kalcsics. 26th European Conference on Operational Research (EURO). Rome, Italy, July 2013.

- *The Multi-period Collection Scheduling Problem with Balancing Constraints.* Cristina Núñez, Elena Fernández, Stefan Nickel, Jörg Kalcsics. 20th Conference of the International Federation of Operational Research Societies (IFORS). Barcelona, Spain, July 2014.

- *Solution algorithms for the multi-period collection Scheduling Problem.* Cristina Núñez, Elena Fernández, Jörg Kalcsics, Stefan Nickel. 17th Latin-Iberian-American Conference on Operations Research (CLAIO). Monterrey, Mexico, October 2014.

- *A column generation algorithm for the multi-period collection scheduling problem.* Cristina Núñez, Elena Fernández, Jörg Kalcsics, Stefan Nickel. 6th International Workshop on Freight Transportation and Logistics (Odysseus). Ajaccio, France, June 2015.

- *Scheduling policies for Multi-period Collection.* Cristina Núñez, Elena Fernández, Jörg Kalcsics, Stefan Nickel. 5th Workshop on Combinatorial Optimization, Routing and Location (CORAL). Salamanca, Spain, September 2015.

Several research opportunities arise from this thesis. One of them appears when two or more entities are involved in the execution of the services. This scenario arises for collaborative systems in which entities have some common interests such as avoiding monopolies. For this scenario, entities may share the utilization of resources, and operative and balancing limitations may apply. A different avenue for research is to consider non-homogeneous capacities. In particular, this scenario appears for operators with different capacities or abilities, and/or customers requiring different types of services. Finally, a clear extension of our study is to consider problems in which the demand of all customers is not the same.

Bibliography

A. Allahverdi, C.T. Ng, T.C.E. Cheng, and M.Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985 – 1032, 2008.

C. Archetti, . Bianchessi, S. Irnich, and M.G. Speranza. Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374, 2014.

A. Atef, M. Abdel-Baset, and I. El-henawy. Project scheduling: Survey and research potentials. *International Jorunal of Computer Applications Technology and Research*, 4(4):235–241, 2015.

P. Avella, M. Boccia, and L.A. Wolsey. Single-item reformulations for a vendor managed inventory routing problem: Computational experience with benchmark instances. *Networks*, 65(2):129–138, 2015.

K.R. Baker. *Principles of Sequencing and Scheduling*. Wiley Publishing, 1974.

K.R. Baker and D. Trietsch. *Principles of Sequencing and Scheduling*. Wiley Publishing, 2009.

A. Bar-Noy and R.E. Ladner. Windows scheduling problems for broadcast systems. *SIAM Journal on Computing*, 32(4):1091–1113, 2003.

A. Bar-Noy, R. Bhatia, J. Naor, and B. Schieber. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27(3):518–544, 2002.

A. Bar-Noy, R.E. Ladner, and T. Tamir. Windows scheduling as a restricted version of bin packing. *ACM Transactions on Algorithms*, 3(3), 2007.

A. Bar-Noy, R.E. Ladner, T. Tamir, and T. Van De Grift. Windows scheduling of arbitrary-length jobs on multiple machines. *Journal of Scheduling*, 15(2):141–155, 2012.

C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

J-C. Billaut, A. Moukrim, and E. Sanlaville. *Flexibility and robustness in scheduling*. John Wiley & Sons, 2013.

D. Bommisetty, M. Dessouky, and L. Jacobs. Scheduling collection of recyclable material at northern illinois university campus using a two-phase algorithm. *Computers & Industrial Engineering*, 35(3): 435–438, 1998.

A. M. Campbell and J. H. Wilson. Forty years of periodic vehicle routing. *Networks*, 63(1):2–15, 2014.

A.M. Campbell and J.R. Hardin. Vehicle minimization for periodic deliveries. *European Journal of Operational Research*, 165(3):668–684, 2005.

B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932, 2010.

J. W.-T. Chan, T.-W. Lam, and P.W.H. Wong. Dynamic bin packing of unit fractions items. *Theoretical Computer Science*, 409(3):521–529, 2008.

I.A. Chaudhry and A.A. Khan. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, pages n/a–n/a, 2015.

A. Cheung, W.H. Ip, D. Lu, and C.L. Lai. An aircraft service scheduling model using genetic algorithms. *Journal of Manufacturing Technology Management*, 16(1):109–119, 2005.

J. Choi, K. Hitomi, P.E. Econ, and P. Econ. Special issue on pacific manufacturing a method of flexible scheduling for flexible manufacturing systems. *International Journal of Production Economics*, 33(1): 247 – 255, 1994.

L.C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565, 2013.

L.C. Coelho, J.F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

E.G. Coffman, Jr, M.R. Garey, and D.S. Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM Journal on Computing*, 7(1):1–17, 1978.

A. Corominas, W. Kubiak, and R. Pastor. Mathematical programming modeling of the response time variability problem. *European Journal of Operational Research*, 200(2):347–357, 2010.

A. Corominas, W. Kubiak, and R. Pastor. Heuristics for the response time variability problem. *European Journal of Industrial Engineering*, 6(6):751–764, 2012.

KA Deshmane and BT Pandhare. Survey on scheduling algorithms in cloud computing. *International Journal of Emerging Trend in Engineering and Basic Sciences*, 2(1):210–2015, 2015.

J.L. Fargeas, B. Hyun, P. Kabamba, and A. Girard. Persistent visitation with fuel constraints. *Procedia-Social and Behavioral Sciences*, 54:1037–1046, 2012.

G. Farkas. A fourier-féle mechanikai elv alkamazásai [hungarian]. *Mathematikai és Természettudományi Értesítö*, 12:457–472, 1894.

D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR*, 8 (4):407–424, 2010.

E. Fernández, J. Kalcsics, S. Nickel, and R.Z. Ríos-Mercado. A novel territory design model arising in the implementation of the WEEE-directive. *Journal of the Operation Research Society*, 61(3):503–514, 2009.

P. Francis and K. Smilowitz. The period vehicle routing problem with services choice. *Transportation Science*, 40(4):439–746, 2006.

G. Gamrath. Generic branch-cut-and-price. Diploma thesis, Institut für Mathematik, Technische Universität Berlin, 2010.

A. García-Villoria, A. Corominas, X. Delorme, A. Dolgui, W. Kubiak, and R. Pastor. A branch and bound algorithm for the response time variability problem. *Journal of Scheduling*, 16(2):243–252, 2013.

M. Gaudioso and G. Paletta. A heuristic for the periodic vehicle routing problem. *Transportation Science*, 26(2):86–92, 1992.

V. Gaur and M.L. Fisher. A periodic inventory routing problem at a supermarket chain. *European Journal of Operational Research*, 52(6):813–822, 2004.

J.V. Gautam, H.B. Prajapati, V.K. Dabhi, and S. Chaudhary. A survey on job scheduling algorithms in big data processing. In *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*, pages 1–11. IEEE, 2015.

L.B. Gonçalves, L.S. Ochi, and S.L. Martins. A grasp with adaptive memory for a period vehicle routing problem. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, volume 1, pages 721–727. IEEE, 2005.

A. Grigoriev, J. van de Klundert, and F.C.R. Spieksma. Modeling and solving the periodic maintenance problem. *European Journal of Operational Research*, 172(3):783–797, 2006.

C.-C. Han, K.-J. Lin, and C.-J. Hou. Distance-constrained scheduling and its applications to real-time systems. *IEEE Transactions on Computers*, 45(7):814–826, 1996.

S. Hartmann and D. Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14, 2010.

J.W. Herrmann. Using aggregation to reduce response time variability in cyclic fair sequences. *Journal of Scheduling*, 14(1):39–55, 2011.

A. Janiak, W.A. Janiak, T. Krysiak, and T. Kwiatkowski. A survey on scheduling problems with due windows. *European Journal of Operational Research*, 242(2):347 – 357, 2015.

T. Jeffries and J. O'Hanley. Optimal location and scheduling of blood collection facilities. In *OR54, University of Edinburgh, 4-6 September 2012*, 2012.

M. Kalra and S. Singh. A review of metaheuristic scheduling techniques in cloud computing. *Egyptian Informatics Journal*, 2015.

W. Karush and A. Vazsonyi. Mathematical programming and service scheduling. *Management Science*, 3(2):140–148, 1957.

M. Khouja and S. Goyal. A review of the joint replenishment problem literature: 1989-2005. *European Journal of Operational Research*, 186(1):1–16, 2008.

A.W.J. Kolen, J.K. Lenstra, C.H. Papadimitriou, and F.C.R. Spieksma. Interval scheduling: A survey. *Naval Research Logistics (NRL)*, 54(5):530–543, 2007.

J. Korst, E. Aarts, J.K. Lenstra, and J. Wessels. Periodic assignment and graph colouring. *Discrete Applied Mathematics*, 51(3):291–305, 1994.

M.Y. Kovalyov, C.T. Ng, and T.C.E. Cheng. Fixed interval scheduling: Models, applications, computational complexity and algorithms. *European Journal of Operational Research*, 178(2):331 – 342, 2007.

V. Lauff and F. Werner. Scheduling with common due date, earliness and tardiness penalties for multi-machine problems: A survey. *Mathematical and Computer Modelling*, 40(5):637 – 655, 2004.

H.M. Le Blanc, F. Cruijssen, H.A. Fleuren, and M.B.M. De Koster. Factory gate pricing: An analysis of the dutch retail distribution. *European Journal of Operational Research*, 174(3):1950–1967, 2006.

S. Liu, G. Quan, and S. Ren. On-line scheduling of real-time services with profit and penalty. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 1476–1481, New York, NY, USA, 2011. ACM.

P. Lopez and F. Roubellat. *Production scheduling*. John Wiley & Sons, 2013.

M.E. Lübbecke. Column generation. In J.J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Chichester, 2011.

A. Mehrotra and M.A. Trick. A branch-and-price approach for graph multi-coloring. In *Extending the horizons: Advances in computing, optimization, and decision technologies*, pages 15–29. Springer, 2007.

J. B. Orlin. Minimizing the number of vehicles to meet a fixed periodic schedule: An application of periodic posets. *Operations Research*, 30(4):760–776, 1982.

K.S. Park and D.K. Yun. Optimal scheduling of periodic activities. *Operations Research*, 33(3):690–695, 1985.

A. Pessoa, E. Uchoa, M. Poggi de Aragão, and R. Rodrigues. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, 2 (3-4):259–290, 2010.

M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer Publishing Company, Incorporated, 2nd edition, 2001.

M. L. Pinedo. *Planning and Scheduling in Manufacturing and Services*. Springer Series in Operations Research. Springer, New York, 2005.

Y. Pochet and L.A. Wolsey. *Production planning by mixed integer programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York, Berlin, 2006.

D. Queiruga, G. Walther, J. González-Benito, and T. Spengler. Evaluation of sites for the location of WEEE recycling plants in spain. *Waste Management*, 28(1):181 – 190, 2008.

T. M. Rossi and J.E. Braun. Minimizing operating costs of vapor compression equipment with optimal service scheduling. *HVAC&R Research*, 2(1):3–25, 1996.

A. Rusdiansyah and D. Tsao. An integrated model of the periodic delivery problems for vending-machine supply chains. *Computers & Operation Research*, 70(3):421–434, 2005.

A. Sbihi and R.W. Eglese. Combinatorial optimization and green logistics. *4OR: A Quarterly Journal of Operation Research*, 5(2):99–116, 2007.

J.B. Sidney. Optimal single-machine scheduling with earliness and tardiness penalties. *Operations Research*, 25(1):62–69, 1977.

O. Solyalı and H. Süral. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Science*, 45(3):335–345, 2011.

F. Vanderbeck. Branching in branch-and-price: a generic scheme. *Mathematical Programming*, 130(2): 249–294, 2011.

A. Violin. *Mathematical programming approaches to pricing problems*. PhD thesis, Universite Libre de Bruxelles, 2014.

O. Wäldrich, P. Wieder, and W. Ziegler. A meta-scheduling service for co-allocating arbitrary types of resources. In Roman Wyrzykowski, Jack Dongarra, Norbert Meyer, and Jerzy WaÅ›niewski, editors, *Parallel Processing and Applied Mathematics*, volume 3911 of *Lecture Notes in Computer Science*, pages 782–791. Springer Berlin Heidelberg, 2006.

P. Wentges. Weighted dantzig-wolfe decomposition for linear mixed-integer programming. *International Transactions in Operational Research*, 4(2):151–162, 1997.

M. Yavuz and D. Jeffcoat. An analysis and solution of the sensor scheduling problem. volume 369, pages 167–177, 2007.

Notation and Abbreviations

Notation

T Set of discretized time periods.

I Set of customers.

t_i service interval for customer $i \in I$ (in number of periods).

Q Capacity of the operators (in number of customers to visit on a given period).

K Set of operators for each period.

β Weight parameter on the importance between number of operators and early periods within services.

C_i Calendar for customer i.

S Schedule for the MSSP (set of periods for which each customer is served). $S = \{C_i : i \in I\}$.

z Total number of operators used in the schedule.

e Total earliness in the schedule.

m_i^t Maximum number of potential periods to schedule a visit for customer $i \in I$ after period $t \in \{0, ..., |T| - 1\}$.

H_i^t Set of potential periods to schedule a visit for customer $i \in I$ after period $t \in \{0, ..., |T| - 1\}$. $H_i^t = \{1, \ldots, m_i^t\}$.

p_i^t Number of potential periods to schedule a visit for customer $i \in I$ before period $t \in T$.

J Set of the different interval classes.

w_j Number of customers of class j.

u_j Service interval of class j.

m_j^t Number of potential periods to schedule a visit for customers of class $j \in J$ after period $t \in \{0, ..., |T| - 1\}$.

H_j^t Set of potential periods to schedule a visit for customers of class $j \in J$ after period $t \in \{0, ..., |T| - 1\}$. $H_j^t = \{1, \ldots, m_j^t\}$.

p_j^t Number of potential periods to schedule a visit for customers of class $j \in J$ before period $t \in T$.

c Pattern of customers to serve in a given period.

a_j^c *j-th* element of pattern c.

n_c Cost of pattern c.

C^t Set of all patterns for visits in period t.

C_i^t Set of patterns for period t containing customer i.

C Set of all possibles patterns in the entire time horizon.

Abbreviations

A-MSSP Aperiodic Multi-Period Service Scheduling Problem.

AS Aperiodic service policy.

BP Branch-and-price.

CG Column Generation.

LP Linear Programming.

MILP Mixed Integer Linear Programming.

MP Master Problem.

MSSP Multi-Period Service Scheduling Problem.

OR Operation Research.

P-MSSP Periodic Multi-Period Service Scheduling Problem.

PS Periodic service policy.

RMP Restricted Master Problem.

List of Figures

List of Tables

List of Algorithms

Appendix A

Results for Formulations for the P-MSSP

Instance		Sparse formulations						Dense formulations					
		PS			PS^c			PD			PD^c		
		z	Gap	$Time$	z	Gap	$Time$	z	Gap	$Time$	z	Gap	$Time$
$I10$	D_I10_Q5_C1	6	0.0	0.47	6	0.0	0.03	6	0.0	0.02	6	0.0	0.02
	D_I10_Q5_C2	12	0.0	0.98	12	0.0	0.25	12	0.0	0.20	12	0.0	0.06
	D_I10_Q5_C3	9	0.0	1.09	9	0.0	0.17	9	0.0	0.09	9	0.0	0.09
	D_I10_Q5_C4	12	0.0	0.86	12	0.0	0.23	12	0.0	0.08	12	0.0	0.09
	D_I10_Q5_C5	12	0.0	0.67	12	0.0	0.34	12	0.0	0.03	12	0.0	0.08
	D_I10_Q10_C1	5	0.0	0.19	5	0.0	0.03	5	0.0	0.25	5	0.0	0.08
	D_I10_Q10_C2	10	0.0	0.08	10	0.0	0.28	10	0.0	0.11	10	0.0	0.27
	D_I10_Q10_C3	5	0.0	0.22	5	0.0	0.06	5	0.0	0.28	5	0.0	0.06
	D_I10_Q10_C4	10	0.0	0.08	10	0.0	0.09	10	0.0	0.11	10	0.0	0.09
	D_I10_Q10_C5	10	0.0	0.13	10	0.0	0.13	10	0.0	0.16	10	0.0	0.09
$I30$	D_I30_Q5_C1	23	0.0	817.12	23	0.0	8.52	23	0.0	2.08	23	0.0	1.01
	D_I30_Q5_C2	29	0.0	1550.42	29	0.0	8.55	29	0.0	1.81	29	0.0	0.94
	D_I30_Q5_C3	28	0.0	1171.81	28	0.0	8.14	28	0.0	1.25	28	0.0	0.94
	D_I30_Q5_C4	30	0.0	2227.84	30	0.0	13.59	30	0.0	7.22	30	0.0	2.56
	D_I30_Q5_C5	30	0.0	2070.85	30	0.0	17.08	30	0.0	3.29	30	0.0	1.62
	D_I30_Q10_C1	14	0.0	4.43	14	0.0	0.59	14	0.0	0.27	14	0.0	0.27
	D_I30_Q10_C2	17	0.0	35.72	17	0.0	0.66	17	0.0	0.34	17	0.0	0.17
	D_I30_Q10_C3	15	0.0	5.80	15	0.0	0.28	15	0.0	0.16	15	0.0	0.06
	D_I30_Q10_C4	18	0.0	47.22	18	0.0	1.12	18	0.0	0.34	18	0.0	0.08
	D_I30_Q10_C5	15	0.0	2.62	15	0.0	0.17	15	0.0	0.09	15	0.0	0.08
	D_I30_Q30_C1	10	0.0	0.27	10	0.0	0.22	10	0.0	0.25	10	0.0	0.31
	D_I30_Q30_C2	10	0.0	0.14	10	0.0	0.22	10	0.0	0.16	10	0.0	0.13
	D_I30_Q30_C3	10	0.0	0.09	10	0.0	0.09	10	0.0	0.14	10	0.0	0.06
	D_I30_Q30_C4	10	0.0	0.11	10	0.0	0.09	10	0.0	0.14	10	0.0	0.05
	D_I30_Q30_C5	10	0.0	0.08	10	0.0	0.08	10	0.0	0.14	10	0.0	0.05
$I50$	D_I50_Q5_C1	45	5.8	3601.72	44	0.0	356.70	44	0.0	923.72	44	0.0	43.98
	D_I50_Q5_C2	47	4.3	3602.29	47	0.0	324.16	47	0.0	852.55	47	0.0	15.09
	D_I50_Q5_C3	49	4.1	3602.05	49	0.0	136.40	49	0.0	12.36	49	0.0	3.71
	D_I50_Q5_C4	50	5.2	3601.86	50	0.0	96.73	50	0.0	24.12	50	0.0	8.66
	D_I50_Q5_C5	52	2.7	3601.78	52	0.0	40.25	52	0.0	23.35	52	0.0	3.48
	D_I50_Q10_C1	25	0.0	1604.57	25	0.0	8.04	25	0.0	5.15	25	0.0	1.70
	D_I50_Q10_C2	24	0.0	433.62	24	0.0	2.34	24	0.0	0.55	24	0.0	0.11
	D_I50_Q10_C3	27	0.0	1047.99	27	0.0	3.54	27	0.0	1.87	27	0.0	0.36
	D_I50_Q10_C4	28	10.0	3600.88	28	0.0	7.88	28	0.0	2.25	28	0.0	1.15
	D_I50_Q10_C5	28	0.0	232.52	28	0.0	2.42	28	0.0	0.55	28	0.0	0.17
	D_I50_Q50_C1	10	0.0	0.12	10	0.0	0.05	10	0.0	0.14	10	0.0	0.11
	D_I50_Q50_C2	10	0.0	0.13	10	0.0	0.11	10	0.0	0.14	10	0.0	0.09
	D_I50_Q50_C3	10	0.0	0.15	10	0.0	0.23	10	0.0	0.14	10	0.0	0.25
	D_I50_Q50_C4	10	0.0	0.14	10	0.0	0.09	10	0.0	0.17	10	0.0	0.03
	D_I50_Q50_C5	10	0.0	0.12	10	0.0	0.09	10	0.0	0.17	10	0.0	0.03
Cumulative		785	32.0	32869.19	784	0.0	1040.04	784	0.0	1866.23	784	0.00	88.17

Table A.1: Detailed CPLEX results for the P-MSSP formulations (D-instances).

Instance		Sparse formulations						Dense formulations					
		PS			PS^c			PD			PD^c		
		z	Gap	$Time$	z	Gap	$Time$	z	Gap	$Time$	z	Gap	$Time$
$I10$	U_I10_Q5_C1	7	0.0	0.55	7	0.0	0.55	7	0.0	0.28	7	0.0	0.11
	U_I10_Q5_C2	13	0.0	0.61	13	0.0	0.98	13	0.0	0.06	13	0.0	0.05
	U_I10_Q5_C3	12	0.0	0.48	12	0.0	0.50	12	0.0	0.05	12	0.0	0.23
	U_I10_Q5_C4	13	0.0	0.46	13	0.0	0.87	13	0.0	0.06	13	0.0	0.19
	U_I10_Q5_C5	10	0.0	0.48	10	0.0	0.36	10	0.0	0.12	10	0.0	0.06
	U_I10_Q10_C1	7	0.0	0.25	7	0.0	0.16	7	0.0	0.20	7	0.0	0.23
	U_I10_Q10_C2	13	0.0	0.09	13	0.0	0.09	13	0.0	0.08	13	0.0	0.09
	U_I10_Q10_C3	12	0.0	0.25	12	0.0	0.28	12	0.0	0.19	12	0.0	0.25
	U_I10_Q10_C4	13	0.0	0.09	13	0.0	0.08	13	0.0	0.09	13	0.0	0.05
	U_I10_Q10_C5	10	0.0	0.19	10	0.0	0.22	10	0.0	0.05	10	0.0	0.11
$I30$	U_I30_Q5_C1	18	0.0	110.99	18	0.0	14.65	18	0.0	1.37	18	0.0	0.69
	U_I30_Q5_C2	22	0.0	855.59	22	0.0	97.53	22	0.0	5.48	22	0.0	4.73
	U_I30_Q5_C3	22	0.0	121.96	22	0.0	40.15	22	0.0	1.72	22	0.0	3.70
	U_I30_Q5_C4	24	0.0	128.36	24	0.0	5.79	24	0.0	0.49	24	0.0	0.09
	U_I30_Q5_C5	23	0.0	404.88	23	0.0	46.69	23	0.0	2.51	23	0.0	1.40
	U_I30_Q10_C1	12	0.0	5.99	12	0.0	3.31	12	0.0	0.17	12	0.0	0.11
	U_I30_Q10_C2	15	0.0	18.03	15	0.0	1.25	15	0.0	0.28	15	0.0	0.16
	U_I30_Q10_C3	14	0.0	11.39	14	0.0	1.36	14	0.0	0.17	14	0.0	0.28
	U_I30_Q10_C4	15	0.0	10.76	15	0.0	1.20	15	0.0	0.27	15	0.0	0.23
	U_I30_Q10_C5	15	0.0	10.08	15	0.0	2.00	15	0.0	0.31	15	0.0	0.47
	U_I30_Q30_C1	12	0.0	0.28	12	0.0	0.33	12	0.0	0.45	12	0.0	0.23
	U_I30_Q30_C2	15	0.0	0.19	15	0.0	0.13	15	0.0	0.47	15	0.0	0.25
	U_I30_Q30_C3	13	0.0	0.22	13	0.0	0.16	13	0.0	0.41	13	0.0	0.09
	U_I30_Q30_C4	15	0.0	0.17	15	0.0	0.22	15	0.0	0.19	15	0.0	0.27
	U_I30_Q30_C5	15	0.0	0.29	15	0.0	0.09	15	0.0	0.19	15	0.0	0.09
$I50$	U_I50_Q5_C1	34	0.0	122.43	34	0.0	25.29	34	0.0	2.42	34	0.0	0.64
	U_I50_Q5_C2	38	0.0	851.36	38	0.0	24.66	38	0.0	1.87	38	0.0	1.25
	U_I50_Q5_C3	38	2.6	3600.12	38	2.6	3600.80	37	0.0	70.67	37	0.0	15.05
	U_I50_Q5_C4	37	2.7	3600.01	37	2.7	3600.23	36	0.0	29.72	36	0.0	39.59
	U_I50_Q5_C5	39	3.1	3602.05	38	0.0	108.02	38	0.0	19.38	38	0.0	3.70
	U_I50_Q10_C1	18	0.0	159.08	18	0.0	14.04	18	0.0	0.72	18	0.0	0.78
	U_I50_Q10_C2	20	0.0	569.65	20	0.0	22.00	20	0.0	3.24	20	0.0	1.39
	U_I50_Q10_C3	20	0.0	226.22	20	0.0	20.33	20	0.0	0.84	20	0.0	0.87
	U_I50_Q10_C4	20	0.0	328.08	20	0.0	18.38	20	0.0	3.48	20	0.0	1.33
	U_I50_Q10_C5	21	0.0	530.73	21	0.0	59.86	21	0.0	4.38	21	0.0	1.78
	U_I50_Q50_C1	15	0.0	0.40	15	0.0	0.30	15	0.0	0.41	15	0.0	0.27
	U_I50_Q50_C2	15	0.0	0.36	15	0.0	0.11	15	0.0	0.48	15	0.0	0.14
	U_I50_Q50_C3	15	0.0	0.39	15	0.0	0.31	15	0.0	0.44	15	0.0	0.16
	U_I50_Q50_C4	15	0.0	0.43	15	0.0	0.11	15	0.0	0.30	15	0.0	0.13
	U_I50_Q50_C5	15	0.0	0.51	15	0.0	0.16	15	0.0	0.52	15	0.0	0.11
Cumulative		720	8.4	15274.42	719	5.3	7713.55	717	0.0	154.49	717	0.00	81.33

Table A.2: Detailed CPLEX results for the P-MSSP formulations (U-instances).

Appendix B

Results for Formulations for the A-MSSP

B.1 Detailed results for formulation *AS*

	Instance	*AS*															
		$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$			
		z	*e*	*Gap*	*Time*	*z*	*e*	*Gap*	*Time*	*z*	*e*	*Gap*	*Time*	*z*	*e*	*Gap*	*Time*
	D_I10_Q5_C1	6	0	0.0	0.2	6	0	0.0	0.3	6	0	0.0	0.3	6	5	0.0	0.3
	D_I10_Q5_C2	12	0	0.0	0.9	12	0	0.0	2.0	8	10	0.0	4.3	8	12	0.0	2.8
	D_I10_Q5_C3	9	0	0.0	1.1	9	0	0.0	1.5	8	1	0.0	5.3	8	60	0.0	42.9
	D_I10_Q5_C4	12	0	0.0	1.0	12	0	0.0	1.8	11	1	0.0	5.4	10	36	0.0	4.6
	D_I10_Q5_C5	12	0	0.0	0.5	12	0	0.0	1.0	11	2	0.0	1.7	11	13	0.0	0.8
I10																	
	D_I10_Q10_C1	5	0	0.0	0.3	5	0	0.0	0.4	5	0	0.0	0.7	4	48	0.0	0.6
	D_I10_Q10_C2	10	0	0.0	0.3	10	0	0.0	0.5	9	2	0.0	1.9	7	77	0.0	2.1
	D_I10_Q10_C3	5	0	0.0	0.3	5	0	0.0	0.3	4	3	0.0	0.5	4	55	0.0	0.3
	D_I10_Q10_C4	10	0	0.0	0.4	10	0	0.0	0.5	9	3	0.0	1.5	7	86	0.0	0.5
	D_I10_Q10_C5	10	0	0.0	0.3	10	0	0.0	0.4	10	0	0.0	2.0	7	76	0.0	0.5
	D_I30_Q5_C1	23	0	0.0	1558.8	23	0	0.0	1676.2	23	0	7.0	3600.8	22	87	0.0	148.4
	D_I30_Q5_C2	29	0	4.1	3601.7	29	0	7.2	3601.5	27	6	7.4	3602.3	27	41	0.0	14.9
	D_I30_Q5_C3	28	0	0.0	2380.1	28	0	7.1	3602.5	26	4	5.2	3602.6	26	84	0.0	220.3
	D_I30_Q5_C4	31	0	9.0	3601.4	29	1	9.3	3602.9	28	4	6.9	3606.7	27	74	0.0	94.7
	D_I30_Q5_C5	30	0	0.0	1920.2	30	0	8.7	3602.5	28	5	6.3	3604.7	28	79	0.0	6.8
	D_I30_Q10_C1	14	0	0.0	12.5	14	0	0.0	30.7	14	0	0.0	578.1	12	94	0.0	1668.7
	D_I30_Q10_C2	17	0	0.0	65.7	17	0	0.0	275.3	15	6	0.0	1624.6	14	72	0.0	355.6
I30	*D_I30_Q10_C3*	15	0	0.0	25.7	15	0	0.0	83.4	15	0	0.0	492.8	14	95	8.3	3605.2
	D_I30_Q10_C4	18	0	0.0	176.2	18	0	0.0	737.9	16	4	11.8	3603.2	15	111	7.6	3600.1
	D_I30_Q10_C5	15	0	0.0	7.4	15	0	0.0	32.7	15	0	0.0	370.4	15	83	8.1	3602.2
	D_I30_Q30_C1	10	0	0.0	1.1	10	0	0.0	3.0	10	0	0.0	9.4	7	322	0.0	5.7
	D_I30_Q30_C2	10	0	0.0	0.8	10	0	0.0	1.6	10	0	0.0	10.1	7	364	0.0	3.7
	D_I30_Q30_C3	10	0	0.0	0.8	10	0	0.0	2.8	10	0	0.0	7.2	7	261	0.0	4.9
	D_I30_Q30_C4	10	0	0.0	0.9	10	0	0.0	1.4	10	0	0.0	7.4	7	252	0.0	3.1
	D_I30_Q30_C5	10	0	0.0	0.9	10	0	0.0	2.3	10	0	0.0	8.3	7	236	0.0	6.3
	D_I50_Q5_C1	45	0	5.8	3601.7	45	0	5.8	3600.6	43	3	3.1	3601.2	43	92	0.0	97.7
	D_I50_Q5_C2	47	0	4.3	3602.2	47	0	4.3	3604.2	46	3	3.7	3601.2	45	94	0.0	653.4
	D_I50_Q5_C3	49	0	4.1	3602.2	49	0	4.1	3603.7	48	3	3.6	3602.6	47	173	0.0	93.8
	D_I50_Q5_C4	50	0	5.2	3600.3	50	0	5.2	3603.9	48	4	3.3	3604.1	48	105	0.0	113.2
	D_I50_Q5_C5	52	0	2.7	3602.3	52	0	2.7	3603.0	51	1	1.3	3601.0	51	106	0.0	116.1
	D_I50_Q10_C1	25	0	7.6	3601.0	25	0	15.2	3601.7	24	1	12.6	3601.1	22	117	0.0	51.1
	D_I50_Q10_C2	24	0	0.0	2481.6	24	0	0.0	1902.9	24	0	6.3	3603.1	23	182	0.0	47.0
I50	*D_I50_Q10_C3*	27	0	0.0	467.6	27	0	11.9	3600.1	25	3	6.2	3603.9	24	111	0.0	221.2
	D_I50_Q10_C4	28	0	15.0	3601.6	26	1	12.2	3602.9	24	6	7.1	3602.7	24	149	0.0	269.3
	D_I50_Q10_C5	28	0	0.0	2877.3	28	0	8.2	3602.0	28	0	9.6	3602.2	26	64	0.0	133.0
	D_I50_Q50_C1	10	0	0.0	1.3	10	0	0.0	3.5	10	0	0.0	13.8	7	633	0.0	12.9
	D_I50_Q50_C2	10	0	0.0	1.4	10	0	0.0	2.6	10	0	0.0	16.6	7	465	0.0	8.8
	D_I50_Q50_C3	10	0	0.0	1.3	10	0	0.0	2.6	10	0	0.0	20.9	7	362	0.0	9.2
	D_I50_Q50_C4	10	0	0.0	2.0	10	0	0.0	2.7	10	0	0.0	14.6	7	391	0.0	9.2
	D_I50_Q50_C5	10	0	0.0	2.0	10	0	0.0	2.0	10	0	0.0	17.6	7	339	0.0	4.6

Table B.1.1: Detailed CPLEX results for formulation *AS* (D-instances).

Instance		AS															
		$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$			
		z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$
I10	U_I10_Q5_C1	7	0	0.0	1.3	7	0	0.0	2.6	6	2	0.0	8.1	5	35	0.0	0.3
	U_I10_Q5_C2	13	0	0.0	1.5	13	0	0.0	4.3	9	9	0.0	12.5	8	37	0.0	9.6
	U_I10_Q5_C3	12	0	0.0	1.3	12	0	0.0	1.8	11	2	0.0	33.8	9	38	0.0	8.7
	U_I10_Q5_C4	13	0	0.0	1.3	13	0	0.0	6.2	10	7	0.0	39.0	9	51	0.0	6.6
	U_I10_Q5_C5	10	0	0.0	0.9	10	0	0.0	1.9	10	0	0.0	8.1	8	36	0.0	2.1
	U_I10_Q10_C1	7	0	0.0	0.6	7	0	0.0	1.1	5	2	0.0	2.7	3	37	0.0	0.2
	U_I10_Q10_C2	13	0	0.0	0.6	13	0	0.0	2.6	10	5	0.0	9.4	7	113	0.0	1.2
	U_I10_Q10_C3	12	0	0.0	0.6	12	0	0.0	1.3	10	4	0.0	4.3	7	72	0.0	1.1
	U_I10_Q10_C4	13	0	0.0	1.0	13	0	0.0	1.7	9	11	0.0	8.2	7	66	0.0	0.9
	U_I10_Q10_C5	10	0	0.0	0.5	10	0	0.0	1.1	9	2	0.0	5.6	6	32	0.0	1.9
I30	U_I30_Q5_C1	18	0	0.0	225.9	18	0	0.0	541.0	18	0	4.1	3600.2	17	63	0.0	5.2
	U_I30_Q5_C2	22	0	0.0	1938.7	22	0	0.0	2098.3	21	1	3.6	3611.2	21	95	0.0	7.3
	U_I30_Q5_C3	22	0	0.0	570.4	22	0	0.0	933.5	22	0	3.6	3603.2	22	118	0.0	24.7
	U_I30_Q5_C4	24	0	0.0	407.9	24	0	0.0	449.2	24	0	3.3	3604.4	24	93	0.0	113.4
	U_I30_Q5_C5	23	0	0.0	251.1	23	0	0.0	1523.5	22	2	3.8	3601.0	22	52	0.0	11.0
	U_I30_Q10_C1	12	0	0.0	28.0	12	0	0.0	79.8	11	2	0.0	861.9	9	78	0.0	120.7
	U_I30_Q10_C2	15	0	0.0	76.8	15	0	0.0	326.3	13	4	0.0	1283.2	11	80	0.0	220.0
	U_I30_Q10_C3	14	0	0.0	37.3	14	0	0.0	284.0	13	2	0.0	1186.4	11	74	0.0	524.6
	U_I30_Q10_C4	15	0	0.0	41.2	15	0	0.0	398.9	13	4	0.0	757.0	12	69	0.0	257.6
	U_I30_Q10_C5	15	0	0.0	67.7	15	0	0.0	214.9	13	5	0.0	1263.7	12	112	0.0	1700.0
	U_I30_Q30_C1	12	0	0.0	4.1	11	1	0.0	54.5	10	4	0.0	254.3	6	511	0.0	6.0
	U_I30_Q30_C2	15	0	0.0	4.9	15	0	0.0	102.7	12	8	0.0	571.0	7	258	0.0	10.4
	U_I30_Q30_C3	13	0	0.0	3.7	13	0	0.0	9.3	12	3	0.0	549.7	7	292	0.0	5.2
	U_I30_Q30_C4	15	0	0.0	3.5	15	0	0.0	12.1	13	4	0.0	278.8	7	211	0.0	7.7
	U_I30_Q30_C5	15	0	0.0	4.2	15	0	0.0	75.5	13	4	0.0	396.5	7	260	0.0	11.0
I50	U_I50_Q5_C1	34	0	0.0	662.5	34	0	0.0	811.6	34	0	2.4	3601.2	34	71	0.0	170.6
	U_I50_Q5_C2	38	0	0.0	1671.9	38	0	0.0	1564.4	38	0	1.6	3600.2	38	108	0.0	57.8
	U_I50_Q5_C3	38	0	2.6	3600.3	38	0	2.6	3600.0	38	0	2.6	3607.1	37	111	0.0	256.9
	U_I50_Q5_C4	37	0	2.7	3600.1	37	0	2.7	3601.0	37	0	2.7	3600.2	36	89	0.0	103.0
	U_I50_Q5_C5	39	0	3.1	3603.7	38	0	0.0	2525.5	38	1	1.2	3600.2	38	67	0.0	126.5
	U_I50_Q10_C1	18	0	0.0	608.7	18	0	0.0	978.5	18	0	0.0	2653.1	17	134	0.0	999.8
	U_I50_Q10_C2	20	0	0.0	1448.7	20	0	0.0	3561.3	20	0	6.5	3600.3	19	106	0.0	890.8
	U_I50_Q10_C3	20	0	0.0	641.5	20	0	0.0	1522.0	20	0	7.5	3600.5	19	118	0.0	271.4
	U_I50_Q10_C4	20	0	0.0	1393.8	20	0	0.0	2345.5	19	2	6.9	3600.4	18	75	0.0	736.6
	U_I50_Q10_C5	21	0	0.0	1327.3	21	0	0.0	2611.0	21	0	9.2	3600.7	19	175	0.0	399.0
	U_I50_Q50_C1	15	0	0.0	10.4	15	0	0.0	38.9	13	5	0.0	1051.1	7	361	0.0	20.1
	U_I50_Q50_C2	15	0	0.0	10.2	15	0	0.0	55.5	15	0	0.0	2467.9	7	415	0.0	23.7
	U_I50_Q50_C3	15	0	0.0	9.3	15	0	0.0	167.3	13	6	0.0	821.5	7	594	0.0	15.1
	U_I50_Q50_C4	15	0	0.0	11.3	15	0	0.0	69.7	14	3	0.0	1821.7	7	422	0.0	14.7
	U_I50_Q50_C5	15	0	0.0	10.3	15	0	0.0	32.7	13	7	0.0	1181.2	7	299	0.0	25.5

Table B.1.2: Detailed CPLEX results for formulation AS (U-instances).

B.2 Detailed results for formulation AS^c

Instance		AS^c															
		$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$			
		z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$
$I10$	$D\_I10\_Q5\_C1$	6	0	0.0	0.1	6	0	0.0	0.0	6	0	0.0	0.0	6	25	0.0	0.1
	$D\_I10\_Q5\_C2$	12	0	0.0	0.7	12	0	0.0	0.5	8	10	0.0	0.9	8	17	0.0	0.6
	$D\_I10\_Q5\_C3$	9	0	0.0	0.4	9	0	0.0	0.5	8	1	0.0	0.3	8	28	0.0	0.5
	$D\_I10\_Q5\_C4$	12	0	0.0	0.5	12	0	0.0	0.3	11	1	0.0	1.8	10	44	0.0	1.9
	$D\_I10\_Q5\_C5$	12	0	0.0	0.3	12	0	0.0	0.2	11	2	0.0	0.8	11	21	0.0	0.9
	$D\_I10\_Q10\_C1$	5	0	0.0	0.1	5	0	0.0	0.2	5	0	0.0	0.2	4	35	0.0	0.3
	$D\_I10\_Q10\_C2$	10	0	0.0	0.6	10	0	0.0	0.2	9	2	0.0	0.6	7	62	0.0	0.4
	$D\_I10\_Q10\_C3$	5	0	0.0	0.2	5	0	0.0	0.2	4	3	0.0	0.2	4	30	0.0	0.1
	$D\_I10\_Q10\_C4$	10	0	0.0	0.2	10	0	0.0	0.3	9	3	0.0	0.6	7	51	0.0	0.3
	$D\_I10\_Q10\_C5$	10	0	0.0	0.1	10	0	0.0	0.5	10	0	0.0	1.2	7	41	0.0	0.2
$I30$	$D\_I30\_Q5\_C1$	23	0	0.0	17.7	23	0	0.0	12.5	23	0	0.0	371.2	22	66	0.0	1.0
	$D\_I30\_Q5\_C2$	29	0	0.0	12.1	29	0	0.0	16.5	28	2	0.0	136.4	27	79	0.0	0.7
	$D\_I30\_Q5\_C3$	28	0	0.0	20.4	28	0	0.0	26.9	26	4	0.0	14.1	26	43	0.0	0.6
	$D\_I30\_Q5\_C4$	30	0	0.0	19.1	29	1	0.0	22.2	28	4	0.0	82.1	27	65	0.0	1.3
	$D\_I30\_Q5\_C5$	30	0	0.0	16.7	30	0	0.0	46.1	28	5	0.0	77.4	28	68	0.0	0.7
	$D\_I30\_Q10\_C1$	14	0	0.0	0.8	14	0	0.0	1.2	14	0	0.0	4.9	12	144	0.0	2.5
	$D\_I30\_Q10\_C2$	17	0	0.0	1.5	17	0	0.0	1.2	15	6	0.0	4.9	14	51	0.0	1.6
	$D\_I30\_Q10\_C3$	15	0	0.0	0.8	15	0	0.0	1.0	15	0	0.0	2.1	14	103	0.0	4.7
	$D\_I30\_Q10\_C4$	18	0	0.0	3.7	18	0	0.0	4.1	16	4	0.0	8.6	15	56	0.0	4.5
	$D\_I30\_Q10\_C5$	15	0	0.0	0.9	15	0	0.0	0.8	15	0	0.0	0.6	15	65	0.0	4.2
	$D\_I30\_Q30\_C1$	10	0	0.0	0.4	10	0	0.0	0.5	10	0	0.0	0.3	7	214	0.0	0.3
	$D\_I30\_Q30\_C2$	10	0	0.0	0.1	10	0	0.0	0.1	10	0	0.0	0.2	7	84	0.0	0.2
	$D\_I30\_Q30\_C3$	10	0	0.0	0.2	10	0	0.0	0.2	10	0	0.0	0.1	7	296	0.0	0.2
	$D\_I30\_Q30\_C4$	10	0	0.0	0.2	10	0	0.0	0.3	10	0	0.0	0.2	7	177	0.0	0.2
	$D\_I30\_Q30\_C5$	10	0	0.0	0.1	10	0	0.0	0.2	10	0	0.0	0.4	7	221	0.0	0.3
$I50$	$D\_I50\_Q5\_C1$	44	0	0.0	1997.2	44	0	0.0	1602.7	43	3	0.0	2620.8	43	95	0.0	1.0
	$D\_I50\_Q5\_C2$	47	0	0.0	1226.2	47	0	0.0	1612.7	46	2	1.2	3600.3	45	135	0.0	1.2
	$D\_I50\_Q5\_C3$	49	0	0.0	180.6	49	0	0.0	227.2	48	2	0.0	1276.9	47	117	0.0	3.0
	$D\_I50\_Q5\_C4$	50	0	0.0	198.1	50	0	0.0	332.3	48	4	0.0	1218.9	48	127	0.0	1.8
	$D\_I50\_Q5\_C5$	52	0	0.0	67.2	52	0	0.0	156.4	51	1	0.0	210.6	51	55	0.0	0.2
	$D\_I50\_Q10\_C1$	25	0	0.0	16.2	25	0	0.0	22.9	24	1	0.0	50.6	22	138	0.0	1.0
	$D\_I50\_Q10\_C2$	24	0	0.0	1.9	24	0	0.0	3.2	24	0	0.0	10.4	23	123	0.0	0.6
	$D\_I50\_Q10\_C3$	27	0	0.0	6.0	27	0	0.0	14.1	25	3	0.0	11.3	24	100	0.0	1.1
	$D\_I50\_Q10\_C4$	28	0	0.0	11.5	26	1	0.0	19.4	24	6	0.0	15.6	24	17	0.0	1.1
	$D\_I50\_Q10\_C5$	28	0	0.0	5.5	28	0	0.0	6.9	28	0	0.0	17.7	26	107	0.0	0.8
	$D\_I50\_Q50\_C1$	10	0	0.0	0.2	10	0	0.0	0.2	10	0	0.0	0.5	7	431	0.0	0.2
	$D\_I50\_Q50\_C2$	10	0	0.0	0.1	10	0	0.0	0.4	10	0	0.0	0.4	7	375	0.0	0.2
	$D\_I50\_Q50\_C3$	10	0	0.0	0.2	10	0	0.0	0.2	10	0	0.0	0.3	7	262	0.0	0.3
	$D\_I50\_Q50\_C4$	10	0	0.0	0.3	10	0	0.0	0.1	10	0	0.0	0.4	7	304	0.0	0.2
	$D\_I50\_Q50\_C5$	10	0	0.0	0.2	10	0	0.0	0.3	10	0	0.0	0.5	7	193	0.0	0.3

Table B.2.1: Detailed CPLEX results for formulation AS^c (D-instances).

	Instance	AS^c															
		$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$			
		z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$	z	e	Gap	$Time$
$I10$	U_I10_Q5_C1	7	0	0.0	0.9	7	0	0.0	0.7	6	2	0.0	1.6	5	25	0.0	1.0
	U_I10_Q5_C2	13	0	0.0	2.1	12	1	0.0	3.7	10	5	0.0	12.8	8	52	0.0	11.7
	U_I10_Q5_C3	12	0	0.0	1.1	12	0	0.0	1.4	11	2	0.0	12.1	9	43	0.0	11.8
	U_I10_Q5_C4	13	0	0.0	1.0	13	0	0.0	3.4	10	7	0.0	16.3	9	46	0.0	9.1
	U_I10_Q5_C5	10	0	0.0	0.6	10	0	0.0	1.5	10	0	0.0	3.0	8	47	0.0	0.5
	U_I10_Q10_C1	7	0	0.0	0.3	7	0	0.0	0.8	5	2	0.0	0.7	3	39	0.0	0.3
	U_I10_Q10_C2	13	0	0.0	1.1	13	0	0.0	2.3	9	9	0.0	4.9	7	88	0.0	0.3
	U_I10_Q10_C3	12	0	0.0	0.8	12	0	0.0	0.8	10	4	0.0	1.8	7	42	0.0	0.8
	U_I10_Q10_C4	13	0	0.0	3.2	12	1	0.0	1.5	9	11	0.0	11.2	7	119	0.0	0.5
	U_I10_Q10_C5	10	0	0.0	0.4	10	0	0.0	0.6	9	2	0.0	1.2	6	122	0.0	0.4
$I30$	U_I30_Q5_C1	18	0	0.0	19.9	18	0	0.0	30.0	18	0	0.0	378.1	17	59	0.0	2.2
	U_I30_Q5_C2	22	0	0.0	227.0	22	0	0.0	357.2	21	1	0.0	721.0	21	83	0.0	3.9
	U_I30_Q5_C3	22	0	0.0	116.3	22	0	0.0	85.8	22	0	0.0	905.5	22	76	0.0	1.4
	U_I30_Q5_C4	24	0	0.0	20.3	24	0	0.0	7.3	24	0	0.0	122.4	24	96	0.0	1.0
	U_I30_Q5_C5	23	0	0.0	76.6	23	0	0.0	71.6	22	2	0.0	396.9	22	81	0.0	8.6
	U_I30_Q10_C1	12	0	0.0	5.6	12	0	0.0	8.2	11	2	0.0	24.8	9	69	0.0	3.8
	U_I30_Q10_C2	15	0	0.0	11.5	14	1	0.0	18.5	13	4	0.0	37.7	11	99	0.0	8.6
	U_I30_Q10_C3	14	0	0.0	7.4	14	0	0.0	5.9	13	2	0.0	25.7	11	65	0.0	4.6
	U_I30_Q10_C4	15	0	0.0	2.7	15	0	0.0	2.3	13	4	0.0	7.9	12	67	0.0	20.3
	U_I30_Q10_C5	15	0	0.0	4.7	15	0	0.0	8.6	13	5	0.0	31.0	12	84	0.0	41.5
	U_I30_Q30_C1	12	0	0.0	1.5	12	0	0.0	1.5	10	4	0.0	13.9	6	230	0.0	1.8
	U_I30_Q30_C2	15	0	0.0	1.3	15	0	0.0	1.6	13	4	0.0	22.0	7	193	0.0	1.6
	U_I30_Q30_C3	13	0	0.0	0.7	13	0	0.0	1.1	12	3	0.0	16.6	7	235	0.0	0.4
	U_I30_Q30_C4	15	0	0.0	0.5	15	0	0.0	1.2	13	4	0.0	6.2	7	345	0.0	0.5
	U_I30_Q30_C5	15	0	0.0	0.7	15	0	0.0	1.8	13	4	0.0	19.3	7	240	0.0	2.5
$I50$	U_I50_Q5_C1	34	0	0.0	14.1	34	0	0.0	41.9	34	0	2.4	3601.0	34	103	0.0	6.8
	U_I50_Q5_C2	38	0	0.0	85.7	38	0	0.0	137.1	38	0	1.6	3600.6	38	102	0.0	3.5
	U_I50_Q5_C3	38	0	2.6	3600.3	38	0	2.6	3600.0	37	0	0.0	631.0	37	156	0.0	11.1
	U_I50_Q5_C4	37	0	2.7	3600.4	37	0	2.7	3600.5	36	0	0.0	473.1	36	78	0.0	15.7
	U_I50_Q5_C5	38	0	0.0	997.8	38	0	0.0	414.9	38	0	0.0	626.8	38	151	0.0	14.6
	U_I50_Q10_C1	18	0	0.0	18.8	18	0	0.0	48.1	18	0	0.0	51.7	17	63	0.0	28.6
	U_I50_Q10_C2	20	0	0.0	69.4	20	0	0.0	115.5	20	0	0.0	186.4	19	123	0.0	13.4
	U_I50_Q10_C3	20	0	0.0	36.1	20	0	0.0	45.6	20	0	0.0	260.8	19	95	0.0	4.0
	U_I50_Q10_C4	20	0	0.0	21.6	20	0	0.0	27.0	19	2	0.0	136.7	18	139	0.0	17.8
	U_I50_Q10_C5	21	0	0.0	91.0	21	0	0.0	96.1	20	2	0.0	369.4	19	93	0.0	17.6
	U_I50_Q50_C1	15	0	0.0	1.4	15	0	0.0	3.3	13	5	0.0	13.7	7	195	0.0	2.0
	U_I50_Q50_C2	15	0	0.0	1.5	15	0	0.0	3.9	15	0	0.0	12.4	7	310	0.0	1.9
	U_I50_Q50_C3	15	0	0.0	2.0	15	0	0.0	3.5	13	6	0.0	9.3	7	634	0.0	1.1
	U_I50_Q50_C4	15	0	0.0	2.0	15	0	0.0	3.9	14	3	0.0	19.7	7	279	0.0	1.5
	U_I50_Q50_C5	15	0	0.0	2.0	15	0	0.0	3.6	13	7	0.0	8.9	7	239	0.0	2.0

Table B.2.2: Detailed CPLEX results for formulation AS^c (U-instances).

Appendix C

Branch-and-Price Results

C.1 Detailed results for the branch-and-price algorithm with formulation MP

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP
$I10$	$Q5$	$C1$	1.20	1.20	1.2	1.2	0.0	0.0	0.2	12.8	1.00	1.02	1.4	1.4	0.0	0.0	1.3	19.4
		$C2$	1.52	1.58	2.4	2.4	0.0	0.0	0.9	16.3	1.40	1.47	2.6	2.6	0.0	0.0	1.5	20.6
		$C3$	1.36	1.36	1.8	1.8	0.0	0.0	1.1	10.9	1.48	1.53	2.4	2.4	0.0	0.0	1.3	25.3
		$C4$	1.72	1.75	2.4	2.4	0.0	0.0	1.0	18.5	1.44	1.52	2.6	2.6	0.0	0.0	1.3	23.8
		$C5$	2.04	2.05	2.4	2.4	0.0	0.0	0.5	18.5	1.56	1.56	2.0	2.0	0.0	0.0	0.9	17.3
	$Q10$	$C1$	0.60	0.80	1.0	1.0	0.0	0.0	0.3	8.2	0.50	0.65	1.4	1.4	0.0	0.0	0.6	14.4
		$C2$	0.76	1.46	2.0	2.0	0.0	0.0	0.3	9.6	0.70	1.47	2.6	2.6	0.0	0.0	0.6	8.1
		$C3$	0.68	0.80	1.0	1.0	0.0	0.0	0.3	24.2	0.74	1.48	2.4	2.4	0.0	0.0	0.6	8.6
		$C4$	0.86	1.46	2.0	2.0	0.0	0.0	0.4	10.8	0.72	1.48	2.6	2.6	0.0	0.0	1.0	7.9
		$C5$	1.02	1.46	2.0	2.0	0.0	0.0	0.3	15.9	0.78	1.20	2.0	2.0	0.0	0.0	0.5	10.5
$I30$	$Q5$	$C1$	4.28	4.28	4.6	4.6	0.0	0.0	1558.8	584.6	3.36	3.36	3.6	3.6	0.0	0.0	225.9	776.7
		$C2$	5.28	5.28	5.8	5.8	4.1	0.0	3601.7	495.7	4.08	4.08	4.4	4.4	0.0	0.0	1938.7	640.2
		$C3$	5.12	5.12	5.6	5.6	0.0	0.0	2380.1	367.4	4.24	4.24	4.4	4.4	0.0	0.0	570.4	593.1
		$C4$	5.40	5.40	6.2	6.0	9.0	0.0	3601.4	381.5	4.64	4.64	4.8	4.8	0.0	0.0	407.9	714.6
		$C5$	5.48	5.48	6.0	6.0	0.0	0.0	1920.2	395.1	4.28	4.28	4.6	4.6	0.0	0.0	251.1	692.6
	$Q10$	$C1$	2.14	2.14	2.8	2.8	0.0	0.0	12.5	655.3	1.68	1.70	2.4	2.4	0.0	0.0	28.0	888.7
		$C2$	2.64	2.64	3.4	3.4	0.0	0.0	65.7	611.5	2.04	2.04	3.0	3.0	0.0	0.0	76.8	742.6
		$C3$	2.56	2.57	3.0	3.0	0.0	0.0	25.7	481.0	2.12	2.12	2.8	2.8	0.0	0.0	37.3	868.5
		$C4$	2.70	2.70	3.6	3.6	0.0	0.0	176.2	561.8	2.32	2.32	3.0	3.0	0.0	0.0	41.2	750.3
		$C5$	2.74	2.75	3.0	3.0	0.0	0.0	7.4	473.2	2.14	2.16	3.0	3.0	0.0	0.0	67.7	787.1
	$Q30$	$C1$	0.71	1.46	2.0	2.0	0.0	0.0	1.1	1063.5	0.56	1.46	2.4	2.4	0.0	0.0	4.1	274.2
		$C2$	0.88	1.46	2.0	2.0	0.0	0.0	0.8	1498.0	0.68	1.46	3.0	3.0	0.0	0.0	4.9	284.3
		$C3$	0.85	1.46	2.0	2.0	0.0	0.0	0.8	1069.8	0.71	1.46	2.6	2.6	0.0	0.0	3.7	556.0
		$C4$	0.90	1.46	2.0	2.0	0.0	0.0	0.9	1574.6	0.77	1.46	3.0	3.0	0.0	0.0	3.5	458.0
		$C5$	0.91	1.46	2.0	2.0	0.0	0.0	0.9	2476.4	0.71	1.48	3.0	3.0	0.0	0.0	4.2	287.3
$I50$	$Q5$	$C1$	8.48	-	9.0	-	5.8	-	3601.7	-	6.64	-	6.8	-	0.0	-	662.5	-
		$C2$	9.00	-	9.4	-	4.3	-	3602.2	-	7.48	-	7.6	-	0.0	-	1671.9	-
		$C3$	9.40	-	9.8	-	4.1	-	3602.2	-	7.40	-	7.6	-	2.6	-	3600.3	-
		$C4$	9.48	-	10.0	-	5.2	-	3600.3	-	7.20	-	7.4	-	2.7	-	3600.1	-
		$C5$	10.12	-	10.4	-	2.7	-	3602.3	-	7.56	-	7.8	-	3.1	-	3603.7	-
	$Q10$	$C1$	4.24	-	5.0	-	7.6	-	3601.0	-	3.32	-	3.6	-	0.0	-	608.7	-
		$C2$	4.50	-	4.8	-	0.0	-	2481.6	-	3.74	-	4.0	-	0.0	-	1448.7	-
		$C3$	4.70	-	5.4	-	0.0	-	467.6	-	3.70	-	4.0	-	0.0	-	641.5	-
		$C4$	4.74	-	5.6	-	15.0	-	3601.6	-	3.60	-	4.0	-	0.0	-	1393.8	-
		$C5$	5.06	-	5.6	-	0.0	-	2877.3	-	3.78	-	4.2	-	0.0	-	1327.3	-
	$Q50$	$C1$	0.85	-	2.0	-	0.0	-	1.3	-	0.66	1.48	3.0	3.0	0.0	102.7	10.4	3600.1
		$C2$	0.90	-	2.0	-	0.0	-	1.4	-	0.75	-	3.0	-	0.0	-	10.2	-
		$C3$	0.94	-	2.0	-	0.0	-	1.3	-	0.74	1.48	3.0	3.0	0.0	102.7	9.3	3601.7
		$C4$	0.95	-	2.0	-	0.0	-	2.0	-	0.72	1.48	3.0	3.0	0.0	102.7	11.3	3601.6
		$C5$	1.01	-	2.0	-	0.0	-	2.0	-	0.76	-	3.0	-	0.0	-	10.3	-

Table C.1.1: Branch-and-price results comparison for formulation MP ($\beta = 0.2$).

Instance			D-instances z_{LP} AS	D-instances z_{LP} MP	D-instances UB AS	D-instances UB MP	D-instances Gap AS	D-instances Gap MP	D-instances $Time$ AS	D-instances $Time$ MP	U-instances z_{LP} AS	U-instances z_{LP} MP	U-instances UB AS	U-instances UB MP	U-instances Gap AS	U-instances Gap MP	U-instances $Time$ AS	U-instances $Time$ MP
$I10$	$Q5$	$C1$	3.00	3.00	3.0	3.0	0.0	0.0	0.3	13.3	2.50	2.54	3.5	3.5	0.0	0.0	2.6	20.7
		$C2$	3.80	3.94	6.0	6.0	0.0	0.0	2.0	15.4	3.50	3.67	6.5	6.5	0.0	0.0	4.3	17.5
		$C3$	3.40	3.40	4.5	4.5	0.0	0.0	1.5	15.8	3.70	3.83	6.0	6.0	0.0	0.0	1.8	17.4
		$C4$	4.30	4.39	6.0	6.0	0.0	0.0	1.8	17.7	3.60	3.80	6.5	6.5	0.0	0.0	6.2	19.9
		$C5$	5.10	5.14	6.0	6.0	0.0	0.0	1.0	15.7	3.90	3.90	5.0	5.0	0.0	0.0	1.9	15.6
	$Q10$	$C1$	1.50	2.00	2.5	2.5	0.0	0.0	0.4	2.9	1.25	1.63	3.5	3.5	0.0	0.0	1.1	16.2
		$C2$	1.90	3.64	5.0	5.0	0.0	0.0	0.5	11.4	1.75	3.67	6.5	6.5	0.0	0.0	2.6	8.0
		$C3$	1.70	2.00	2.5	2.5	0.0	0.0	0.3	27.3	1.85	3.70	6.0	6.0	0.0	0.0	1.3	8.8
		$C4$	2.15	3.64	5.0	5.0	0.0	0.0	0.5	12.6	1.80	3.70	6.5	6.5	0.0	0.0	1.7	7.8
		$C5$	2.55	3.64	5.0	5.0	0.0	0.0	0.4	18.8	1.95	3.00	5.0	5.0	0.0	0.0	1.1	10.3
$I30$	$Q5$	$C1$	10.70	10.70	11.5	11.5	0.0	0.0	1676.2	1349.2	8.40	8.40	9.0	9.0	0.0	0.0	541.0	1965.2
		$C2$	13.20	13.20	14.5	14.5	7.2	0.0	3601.5	901.9	10.20	10.20	11.0	11.0	0.0	0.0	2098.3	1396.7
		$C3$	12.80	12.80	14.0	14.0	7.1	0.0	3602.5	810.3	10.60	10.60	11.0	11.0	0.0	0.0	933.5	1487.8
		$C4$	13.50	13.50	15.0	15.0	9.3	0.0	3602.9	714.4	11.60	11.60	12.0	12.0	0.0	0.0	449.2	1841.4
		$C5$	13.70	13.70	15.0	15.0	8.7	0.0	3602.5	771.7	10.70	10.70	11.5	11.5	0.0	0.0	1523.5	1699.6
	$Q10$	$C1$	5.35	5.35	7.0	7.0	0.0	0.0	30.7	1171.2	4.20	4.25	6.0	6.0	0.0	0.0	79.8	1687.3
		$C2$	6.60	6.60	8.5	8.5	0.0	0.0	275.3	896.7	5.10	5.11	7.5	7.5	0.0	0.0	326.3	1341.7
		$C3$	6.40	6.42	7.5	7.5	0.0	0.0	83.4	1212.3	5.30	5.30	7.0	7.0	0.0	0.0	284.0	1501.4
		$C4$	6.75	6.75	9.0	9.0	0.0	0.0	737.9	1334.3	5.80	5.80	7.5	7.5	0.0	0.0	398.9	1281.0
		$C5$	6.85	6.88	7.5	7.5	0.0	0.0	32.7	1477.4	5.35	5.40	7.5	7.5	0.0	0.0	214.9	1030.3
	$Q30$	$C1$	1.78	3.64	5.0	5.0	0.0	0.0	3.0	2650.9	1.40	3.00	6.0	6.0	0.0	0.0	54.5	677.3
		$C2$	2.20	3.64	5.0	5.0	0.0	0.0	1.6	3591.0	1.70	3.70	7.5	7.5	0.0	0.0	102.7	788.8
		$C3$	2.13	3.64	5.0	5.0	0.0	37.3	2.8	3600.0	1.77	3.67	6.5	6.5	0.0	0.0	9.3	1093.5
		$C4$	2.25	3.64	5.0	5.0	0.0	37.3	1.4	3600.0	1.93	3.70	7.5	7.5	0.0	0.0	12.1	1182.1
		$C5$	2.28	3.64	5.0	5.0	0.0	37.3	2.3	3600.0	1.78	3.70	7.5	7.5	0.0	0.0	75.5	900.7
$I50$	$Q5$	$C1$	21.20	-	22.5	-	5.8	-	3600.6	-	16.60	-	17.0	-	0.0	-	811.6	-
		$C2$	22.50	-	23.5	-	4.3	-	3604.2	-	18.70	-	19.0	-	0.0	-	1564.4	-
		$C3$	23.50	-	24.5	-	4.1	-	3603.7	-	18.50	-	19.0	-	2.6	-	3600.0	-
		$C4$	23.70	-	25.0	-	5.2	-	3603.9	-	18.00	-	18.5	-	2.7	-	3601.0	-
		$C5$	25.30	-	26.0	-	2.7	-	3603.0	-	18.90	-	19.0	-	0.0	-	2525.5	-
	$Q10$	$C1$	10.60	-	12.5	-	15.2	-	3601.7	-	8.30	-	9.0	-	0.0	-	978.5	-
		$C2$	11.25	-	12.0	-	0.0	-	1902.9	-	9.35	-	10.0	-	0.0	-	3561.3	-
		$C3$	11.75	-	13.5	-	11.9	-	3600.1	-	9.25	-	10.0	-	0.0	-	1522.0	-
		$C4$	11.85	-	13.5	-	12.2	-	3602.9	-	9.00	-	10.0	-	0.0	-	2345.5	-
		$C5$	12.65	-	14.0	-	8.2	-	3602.0	-	9.45	-	10.5	-	0.0	-	2611.0	-
	$Q50$	$C1$	2.12	-	5.0	-	0.0	-	3.5	-	1.66	3.70	7.5	7.5	0.0	102.7	38.9	3600.2
		$C2$	2.25	-	5.0	-	0.0	-	2.6	-	1.87	-	7.5	-	0.0	-	55.5	-
		$C3$	2.35	-	5.0	-	0.0	-	2.6	-	1.85	-	7.5	-	0.0	-	167.3	-
		$C4$	2.37	-	5.0	-	0.0	-	2.7	-	1.80	-	7.5	-	0.0	-	69.7	-
		$C5$	2.53	-	5.0	-	0.0	-	2.0	-	1.89	-	7.5	-	0.0	-	32.7	-

Table C.1.2: Branch-and-price results comparison for formulation MP ($\beta = 0.5$).

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP
$I10$	$Q5$	$C1$	4.80	4.80	4.8	4.8	0.0	0.0	0.3	14.4	4.00	4.06	5.2	5.2	0.0	0.0	8.1	16.7
		$C2$	6.08	6.31	8.4	8.4	0.0	0.0	4.3	16.6	5.60	5.88	9.0	9.0	0.0	0.0	12.5	14.9
		$C3$	5.44	5.44	6.6	6.6	0.0	0.0	5.3	19.7	5.92	6.12	9.2	9.2	0.0	0.0	33.8	15.8
		$C4$	6.88	7.02	9.0	9.0	0.0	0.0	5.4	21.8	5.76	6.09	9.4	9.4	0.0	0.0	39.0	14.7
		$C5$	8.16	8.22	9.2	9.2	0.0	0.0	1.7	16.8	6.24	6.24	8.0	8.0	0.0	0.0	8.1	12.9
	$Q10$	$C1$	2.40	3.20	4.0	4.0	0.0	0.0	0.7	31.3	2.00	2.60	4.4	4.4	0.0	0.0	2.7	12.2
		$C2$	3.04	5.83	7.6	7.6	0.0	0.0	1.9	11.6	2.80	5.87	9.0	9.0	0.0	0.0	9.4	8.3
		$C3$	2.72	3.20	3.8	3.8	0.0	0.0	0.5	47.0	2.96	5.92	8.8	8.8	0.0	0.0	4.3	7.4
		$C4$	3.44	5.83	7.8	7.8	0.0	0.0	1.5	16.6	2.88	5.92	9.4	9.4	0.0	0.0	8.2	7.5
		$C5$	4.08	5.83	8.0	8.0	0.0	0.0	2.0	21.8	3.12	4.80	7.6	7.6	0.0	0.0	5.6	10.8
$I30$	$Q5$	$C1$	17.12	17.12	18.4	18.4	7.0	0.0	3600.8	1934.3	13.44	13.44	14.4	14.4	4.1	0.0	3600.2	1280.3
		$C2$	21.12	21.12	22.8	22.8	7.4	0.0	3602.3	1318.9	16.32	16.32	17.0	17.0	3.6	0.0	3611.2	1613.7
		$C3$	20.48	20.48	21.6	21.6	5.2	0.0	3602.6	1294.0	16.96	16.96	17.6	17.6	3.6	0.0	3603.2	1641.5
		$C4$	21.60	21.60	23.2	23.2	6.9	0.0	3606.7	946.8	18.56	18.56	19.2	19.2	3.3	0.0	3604.4	1895.7
		$C5$	21.92	21.92	23.4	23.4	6.3	0.0	3604.7	1036.0	17.12	17.12	18.0	18.0	3.8	0.0	3601.0	2136.4
	$Q10$	$C1$	8.56	8.56	11.2	11.2	0.0	0.0	578.1	997.8	6.72	6.80	9.2	9.2	0.0	0.0	861.9	1198.0
		$C2$	10.56	10.56	13.2	13.2	0.0	0.0	1624.6	1206.5	8.16	8.18	11.2	11.2	0.0	0.0	1283.2	935.3
		$C3$	10.24	10.27	12.0	12.0	0.0	0.0	492.8	973.3	8.48	8.48	10.8	10.8	0.0	0.0	1186.4	1004.6
		$C4$	10.80	10.80	13.6	13.6	11.8	0.0	3603.2	1056.6	9.28	9.28	11.2	11.2	0.0	0.0	757.0	820.0
		$C5$	10.96	11.00	12.0	12.0	0.0	0.0	370.4	980.0	8.56	8.63	11.4	11.4	0.0	0.0	1263.7	974.1
	$Q30$	$C1$	2.85	5.83	8.0	8.0	0.0	37.3	9.4	3600.0	2.24	4.80	8.8	8.8	0.0	0.0	254.3	537.8
		$C2$	3.52	5.83	8.0	8.0	0.0	37.3	10.1	3600.0	2.72	5.92	11.2	11.2	0.0	0.0	571.0	608.0
		$C3$	3.41	5.83	8.0	8.0	0.0	37.3	7.2	3600.0	2.83	5.87	10.2	10.2	0.0	0.0	549.7	828.7
		$C4$	3.60	5.83	8.0	8.0	0.0	37.3	7.4	3600.0	3.09	5.92	11.2	11.2	0.0	0.0	278.8	638.8
		$C5$	3.65	5.83	8.0	8.0	0.0	37.3	8.3	3600.0	2.85	5.92	11.2	11.2	0.0	0.0	396.5	390.1
$I50$	$Q5$	$C1$	33.92	-	35.0	-	3.1	-	3601.2	-	26.56	-	27.2	-	2.4	-	3601.2	-
		$C2$	36.00	-	37.4	-	3.7	-	3601.2	-	29.92	-	30.4	-	1.6	-	3600.2	-
		$C3$	37.60	-	39.0	-	3.6	-	3602.6	-	29.60	-	30.4	-	2.6	-	3607.1	-
		$C4$	37.92	-	39.2	-	3.3	-	3604.1	-	28.80	-	29.6	-	2.7	-	3600.2	-
		$C5$	40.48	-	41.0	-	1.3	-	3601.0	-	30.24	-	30.6	-	1.2	-	3600.2	-
	$Q10$	$C1$	16.96	-	19.4	-	12.6	-	3601.1	-	13.28	-	14.4	-	0.0	-	2653.1	-
		$C2$	18.00	-	19.2	-	6.3	-	3603.1	-	14.96	-	16.0	-	6.5	-	3600.3	-
		$C3$	18.80	-	20.6	-	6.2	-	3603.9	-	14.80	-	16.0	-	7.5	-	3600.5	-
		$C4$	18.96	-	20.4	-	7.1	-	3602.7	-	14.40	-	15.6	-	6.9	-	3600.4	-
		$C5$	20.24	-	22.4	-	9.6	-	3602.2	-	15.12	-	16.8	-	9.2	-	3600.7	-
	$Q50$	$C1$	3.39	-	8.0	-	0.0	-	13.8	-	2.66	5.92	11.4	11.4	0.0	92.6	1051.1	3601.4
		$C2$	3.60	-	8.0	-	0.0	-	16.6	-	2.99	5.92	12.0	12.0	0.0	102.7	2467.9	3600.1
		$C3$	3.76	-	8.0	-	0.0	-	20.9	-	2.96	5.92	11.6	11.6	0.0	96.0	821.5	3601.6
		$C4$	3.79	-	8.0	-	0.0	-	14.6	-	2.88	5.92	11.8	11.8	0.0	99.3	1821.7	3601.4
		$C5$	4.05	-	8.0	-	0.0	-	17.6	-	3.02	-	11.8	-	0.0	-	1181.2	-

Table C.1.3: Branch-and-price results comparison for formulation MP ($\beta = 0.8$).

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP	AS	MP
$I10$		$C1$	6.00	6.00	6	6	0.0	0.0	0.3	5.5	5.00	5.00	5	5	0.0	0.0	0.3	4.5
		$C2$	7.60	8.00	8	8	0.0	0.0	2.8	8.6	7.00	8.00	8	8	0.0	0.0	9.6	7.3
	$Q5$	$C3$	6.80	6.80	8	8	0.0	0.0	42.9	6.4	7.40	7.61	9	9	0.0	0.0	8.7	11.4
		$C4$	8.60	8.60	10	10	0.0	0.0	4.6	9.5	7.20	7.55	9	9	0.0	0.0	6.6	9.3
		$C5$	10.20	11.00	11	11	0.0	0.0	0.8	7.5	7.80	8.00	8	8	0.0	0.0	2.1	5.3
		$C1$	3.00	4.00	4	4	0.0	0.0	0.6	0.5	2.50	3.00	3	3	0.0	0.0	0.2	0.4
		$C2$	3.80	7.00	7	7	0.0	0.0	2.1	0.6	3.50	7.00	7	7	0.0	0.0	1.2	0.5
	$Q10$	$C3$	3.40	4.00	4	4	0.0	0.0	0.3	0.6	3.70	7.00	7	7	0.0	0.0	1.1	0.6
		$C4$	4.30	7.00	7	7	0.0	0.0	0.5	0.6	3.60	7.00	7	7	0.0	0.0	0.9	0.4
		$C5$	5.10	7.00	7	7	0.0	0.0	0.5	0.5	3.90	6.00	6	6	0.0	0.0	1.9	0.6
		$C1$	21.40	22.00	22	22	0.0	0.0	148.4	62.6	16.80	17.00	17	17	0.0	0.0	5.2	811.6
		$C2$	26.40	27.00	27	27	0.0	0.0	14.9	145.4	20.40	21.00	21	21	0.0	0.0	7.3	232.8
	$Q5$	$C3$	25.60	26.00	26	26	0.0	0.0	220.3	228.1	21.20	22.00	22	22	0.0	0.0	24.7	56.3
		$C4$	27.00	27.00	27	27	0.0	0.0	94.7	628.4	23.20	24.00	24	24	0.0	0.0	113.4	159.4
		$C5$	27.40	28.00	28	28	0.0	0.0	6.8	173.1	21.40	22.00	22	22	0.0	0.0	11.0	614.2
		$C1$	10.70	10.70	12	12	0.0	0.0	1668.7	106.5	8.40	9.00	9	9	0.0	0.0	120.7	411.0
		$C2$	13.20	14.00	14	14	0.0	0.0	355.6	172.8	10.20	11.00	11	11	0.0	0.0	220.0	160.3
$I30$	$Q10$	$C3$	12.80	12.80	14	14	8.3	0.0	3605.2	187.7	10.60	11.00	11	11	0.0	0.0	524.6	257.5
		$C4$	13.50	13.50	15	15	7.6	0.0	3600.1	172.6	11.60	12.00	12	12	0.0	0.0	257.6	303.2
		$C5$	13.70	13.70	15	15	8.1	0.0	3602.2	233.7	10.70	10.70	12	12	0.0	0.0	1700.0	190.9
		$C1$	3.57	7.00	7	7	0.0	0.0	5.7	1.2	2.80	6.00	6	6	0.0	0.0	6.0	1.3
		$C2$	4.40	7.00	7	7	0.0	0.0	3.7	1.3	3.40	7.00	7	7	0.0	0.0	10.4	1.1
	$Q30$	$C3$	4.27	7.00	7	7	0.0	0.0	4.9	0.9	3.53	7.00	7	7	0.0	0.0	5.2	1.1
		$C4$	4.50	7.00	7	7	0.0	0.0	3.1	1.1	3.87	7.00	7	7	0.0	0.0	7.7	1.1
		$C5$	4.57	7.00	7	7	0.0	0.0	6.3	0.9	3.57	7.00	7	7	0.0	0.0	11.0	1.4
		$C1$	42.40	-	43	-	0.0	-	97.7	-	33.20	34.00	34	34	0.0	0.0	170.6	1360.7
		$C2$	45.00	-	45	-	0.0	-	653.4	-	37.40	-	38	-	0.0	-	57.8	-
	$Q5$	$C3$	47.00	-	47	-	0.0	-	93.8	-	37.00	-	37	-	0.0	-	256.9	-
		$C4$	47.40	-	48	-	0.0	-	113.2	-	36.00	-	36	-	0.0	-	103.0	-
		$C5$	50.60	-	51	-	0.0	-	116.1	-	37.80	-	38	-	0.0	-	126.5	-
		$C1$	21.20	22.00	22	22	0.0	0.0	51.1	2352.4	16.60	-	17	-	0.0	-	999.8	-
		$C2$	22.50	-	23	-	0.0	-	47.0	-	18.70	19.00	19	19	0.0	0.0	890.8	3592.5
$I50$	$Q10$	$C3$	23.50	-	24	-	0.0	-	221.2	-	18.50	19.00	19	19	0.0	0.0	271.4	3511.5
		$C4$	23.70	-	24	-	0.0	-	269.3	-	18.00	-	18	-	0.0	-	736.6	-
		$C5$	25.30	-	26	-	0.0	-	133.0	-	18.90	-	19	-	0.0	-	399.0	-
		$C1$	4.24	7.00	7	7	0.0	0.0	12.9	1.3	3.32	7.00	7	7	0.0	0.0	20.1	2.1
		$C2$	4.50	7.00	7	7	0.0	0.0	8.8	2.4	3.74	7.00	7	7	0.0	0.0	23.7	2.0
	$Q50$	$C3$	4.70	7.00	7	7	0.0	0.0	9.2	1.3	3.70	7.00	7	7	0.0	0.0	15.1	1.6
		$C4$	4.74	7.00	7	7	0.0	0.0	9.2	1.5	3.60	7.00	7	7	0.0	0.0	14.7	1.6
		$C5$	5.06	7.00	7	7	0.0	0.0	4.6	2.0	3.78	7.00	7	7	0.0	0.0	25.5	2.2

Table C.1.4: Branch-and-price results comparison for formulation MP ($\beta = 1.0$).

C.2 Detailed results for the branch-and-price algorithm with formulation MP^c

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP
$I10$	$Q5$	$C1$	1.20	1.20	1.2	1.2	0.0	0.0	0.1	0.0	1.00	1.02	1.4	1.4	0.0	0.0	0.9	5.8
		$C2$	1.52	1.58	2.4	2.4	0.0	0.0	0.7	3.3	1.40	1.47	2.6	2.6	0.0	0.0	2.1	32.4
		$C3$	1.36	1.36	1.8	1.8	0.0	0.0	0.4	1.0	1.48	1.53	2.4	2.4	0.0	0.0	1.1	14.0
		$C4$	1.72	1.75	2.4	2.4	0.0	0.0	0.5	2.1	1.44	1.52	2.6	2.6	0.0	0.0	1.0	33.5
		$C5$	2.04	2.05	2.4	2.4	0.0	0.0	0.3	0.5	1.56	1.56	2.0	2.0	0.0	0.0	0.6	3.5
	$Q10$	$C1$	0.60	1.00	1.0	1.0	0.0	0.0	0.1	0.1	0.50	0.65	1.4	1.4	0.0	0.0	0.3	12.0
		$C2$	0.76	1.46	2.0	2.0	0.0	0.0	0.6	1.3	0.70	1.47	2.6	2.6	0.0	0.0	1.1	31.8
		$C3$	0.68	1.00	1.0	1.0	0.0	0.0	0.2	0.1	0.74	1.48	2.4	2.4	0.0	0.0	0.8	12.6
		$C4$	0.86	1.46	2.0	2.0	0.0	0.0	0.2	1.1	0.72	1.48	2.6	2.6	0.0	0.0	3.2	22.5
		$C5$	1.02	1.46	2.0	2.0	0.0	0.0	0.1	1.0	0.78	1.20	2.0	2.0	0.0	0.0	0.4	7.9
$I30$	$Q5$	$C1$	4.28	4.28	4.6	4.6	0.0	0.0	17.7	22.0	3.36	3.36	3.6	3.6	0.0	0.0	19.9	112.3
		$C2$	5.28	5.28	5.8	5.8	0.0	0.0	12.1	38.1	4.08	4.08	4.4	4.4	0.0	0.0	227.0	652.2
		$C3$	5.12	5.12	5.6	5.6	0.0	0.0	20.4	79.0	4.24	4.24	4.4	4.4	0.0	0.0	116.3	50.4
		$C4$	5.40	5.40	6.0	6.0	0.0	0.0	19.1	41.2	4.64	4.64	4.8	4.8	0.0	0.0	20.3	18.7
		$C5$	5.48	5.48	6.0	6.0	0.0	0.0	16.7	157.3	4.28	4.28	4.6	4.6	0.0	0.0	76.6	156.1
	$Q10$	$C1$	2.14	2.14	2.8	2.8	0.0	0.0	0.8	6.2	1.68	1.70	2.4	2.4	0.0	0.0	5.6	25.7
		$C2$	2.64	2.64	3.4	3.4	0.0	0.0	1.5	8.5	2.04	2.04	3.0	3.0	0.0	0.0	11.5	43.5
		$C3$	2.56	2.57	3.0	3.0	0.0	0.0	0.8	4.7	2.12	2.12	2.8	2.8	0.0	0.0	7.4	35.6
		$C4$	2.70	2.70	3.6	3.6	0.0	0.0	3.7	12.9	2.32	2.32	3.0	3.0	0.0	0.0	2.7	25.2
		$C5$	2.74	2.75	3.0	3.0	0.0	0.0	0.9	1.1	2.14	2.16	3.0	3.0	0.0	0.0	4.7	42.8
	$Q30$	$C1$	0.71	1.46	2.0	2.0	0.0	0.0	0.4	1.1	0.56	1.20	2.4	2.4	0.0	0.0	1.5	50.2
		$C2$	0.88	1.46	2.0	2.0	0.0	0.0	0.1	1.2	0.68	1.48	3.0	3.0	0.0	0.0	1.3	82.9
		$C3$	0.85	1.46	2.0	2.0	0.0	0.0	0.2	0.8	0.71	1.47	2.6	2.6	0.0	0.0	0.7	42.6
		$C4$	0.90	1.46	2.0	2.0	0.0	0.0	0.2	0.9	0.77	1.48	3.0	3.0	0.0	0.0	0.5	39.6
		$C5$	0.91	1.46	2.0	2.0	0.0	0.0	0.1	1.0	0.71	1.48	3.0	3.0	0.0	0.0	0.7	59.9
$I50$	$Q5$	$C1$	8.48	8.48	8.8	8.8	0.0	0.0	1997.2	10.3	6.64	6.64	6.8	6.8	0.0	0.0	14.1	118.8
		$C2$	9.00	9.00	9.4	9.4	0.0	0.0	1226.2	17.6	7.48	7.48	7.6	7.6	0.0	0.0	85.7	411.4
		$C3$	9.40	9.40	9.8	9.8	0.0	0.0	180.6	9.1	7.40	7.40	7.6	7.6	2.6	2.7	3600.3	3600.0
		$C4$	9.48	9.48	10.0	10.0	0.0	0.0	198.1	42.7	7.20	7.20	7.4	7.4	2.7	2.8	3600.4	3600.0
		$C5$	10.12	10.12	10.4	10.4	0.0	0.0	67.2	29.5	7.56	7.56	7.6	7.6	0.0	0.0	997.8	2868.7
	$Q10$	$C1$	4.24	4.24	5.0	5.0	0.0	0.0	16.2	152.1	3.32	3.32	3.6	3.6	0.0	0.0	18.8	72.8
		$C2$	4.50	4.50	4.8	4.8	0.0	0.0	1.9	15.6	3.74	3.74	4.0	4.0	0.0	0.0	69.4	128.6
		$C3$	4.70	4.70	5.4	5.4	0.0	0.0	6.0	31.0	3.70	3.70	4.0	4.0	0.0	0.0	36.1	237.2
		$C4$	4.74	4.74	5.6	5.6	0.0	0.0	11.5	100.0	3.60	3.60	4.0	4.0	0.0	0.0	21.6	311.7
		$C5$	5.06	5.06	5.6	5.6	0.0	0.0	5.5	23.9	3.78	3.78	4.2	4.2	0.0	0.0	91.0	354.0
	$Q50$	$C1$	0.85	1.46	2.0	2.0	0.0	0.0	0.2	1.4	0.66	1.48	3.0	3.0	0.0	0.0	1.4	62.2
		$C2$	0.90	1.46	2.0	2.0	0.0	0.0	0.1	1.2	0.75	1.48	3.0	3.0	0.0	0.0	1.5	56.6
		$C3$	0.94	1.46	2.0	2.0	0.0	0.0	0.2	1.1	0.74	1.48	3.0	3.0	0.0	0.0	2.0	53.2
		$C4$	0.95	1.46	2.0	2.0	0.0	0.0	0.3	0.9	0.72	1.48	3.0	3.0	0.0	0.0	2.0	93.9
		$C5$	1.01	1.46	2.0	2.0	0.0	0.0	0.2	1.0	0.76	1.48	3.0	3.0	0.0	0.0	2.0	42.3

Table C.2.1: Branch-and-price results comparison for formulation MP^c ($\beta = 0.2$).

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP
$I10$	$Q5$	$C1$	3.00	3.00	3.0	3.0	0.0	0.0	0.0	0.0	2.50	2.54	3.5	3.5	0.0	0.0	0.7	8.3
		$C2$	3.80	3.94	6.0	6.0	0.0	0.0	0.5	3.8	3.50	3.67	6.5	6.5	0.0	0.0	3.7	157.8
		$C3$	3.40	3.40	4.5	4.5	0.0	0.0	0.5	0.9	3.70	3.83	6.0	6.0	0.0	0.0	1.4	32.5
		$C4$	4.30	4.39	6.0	6.0	0.0	0.0	0.3	2.4	3.60	3.80	6.5	6.5	0.0	0.0	3.4	99.0
		$C5$	5.10	5.14	6.0	6.0	0.0	0.0	0.2	0.5	3.90	3.90	5.0	5.0	0.0	0.0	1.5	4.5
	$Q10$	$C1$	1.50	2.50	2.5	2.5	0.0	0.0	0.2	0.1	1.25	1.63	3.5	3.5	0.0	0.0	0.8	19.6
		$C2$	1.90	3.64	5.0	5.0	0.0	0.0	0.2	4.0	1.75	3.67	6.5	6.5	0.0	0.0	2.3	267.6
		$C3$	1.70	2.50	2.5	2.5	0.0	0.0	0.2	0.1	1.85	3.70	6.0	6.0	0.0	0.0	0.8	75.3
		$C4$	2.15	3.64	5.0	5.0	0.0	0.0	0.3	2.3	1.80	3.70	6.5	6.5	0.0	0.0	1.5	226.6
		$C5$	2.55	3.64	5.0	5.0	0.0	0.0	0.5	1.3	1.95	3.00	5.0	5.0	0.0	0.0	0.6	16.7
$I30$	$Q5$	$C1$	10.70	10.70	11.5	11.5	0.0	0.0	12.5	22.0	8.40	8.40	9.0	9.0	0.0	0.0	30.0	101.9
		$C2$	13.20	13.20	14.5	14.5	0.0	0.0	16.5	38.1	10.20	10.20	11.0	11.0	0.0	0.0	357.2	629.2
		$C3$	12.80	12.80	14.0	14.0	0.0	0.0	26.9	158.2	10.60	10.60	11.0	11.0	0.0	0.0	85.8	23.5
		$C4$	13.50	13.50	15.0	15.0	0.0	0.0	22.2	41.2	11.60	11.60	12.0	12.0	0.0	0.0	7.3	27.9
		$C5$	13.70	13.70	15.0	15.0	0.0	0.0	46.1	163.3	10.70	10.70	11.5	11.5	0.0	0.0	71.6	140.0
	$Q10$	$C1$	5.35	5.35	7.0	7.0	0.0	0.0	1.2	10.1	4.20	4.25	6.0	6.0	0.0	0.0	8.2	54.6
		$C2$	6.60	6.60	8.5	8.5	0.0	0.0	1.2	8.4	5.10	5.11	7.5	7.5	0.0	0.0	18.5	137.5
		$C3$	6.40	6.42	7.5	7.5	0.0	0.0	1.0	3.6	5.30	5.30	7.0	7.0	0.0	0.0	5.9	41.1
		$C4$	6.75	6.75	9.0	9.0	0.0	0.0	4.1	31.5	5.80	5.80	7.5	7.5	0.0	0.0	2.3	28.0
		$C5$	6.85	6.88	7.5	7.5	0.0	0.0	0.8	2.1	5.35	5.40	7.5	7.5	0.0	0.0	8.6	64.1
	$Q30$	$C1$	1.78	3.64	5.0	5.0	0.0	0.0	0.5	4.2	1.40	3.00	6.0	6.0	0.0	0.0	1.5	414.7
		$C2$	2.20	3.64	5.0	5.0	0.0	0.0	0.1	1.4	1.70	3.70	7.5	7.5	0.0	0.0	1.6	570.6
		$C3$	2.13	3.64	5.0	5.0	0.0	0.0	0.2	1.7	1.77	3.67	6.5	6.5	0.0	0.0	1.1	107.6
		$C4$	2.25	3.64	5.0	5.0	0.0	0.0	0.3	1.6	1.93	3.70	7.5	7.5	0.0	0.0	1.2	76.6
		$C5$	2.28	3.64	5.0	5.0	0.0	0.0	0.2	1.1	1.78	3.70	7.5	7.5	0.0	0.0	1.8	271.9
$I50$	$Q5$	$C1$	21.20	21.20	22.0	22.0	0.0	0.0	1602.7	10.3	16.60	16.60	17.0	17.0	0.0	0.0	41.9	170.9
		$C2$	22.50	22.50	23.5	23.5	0.0	0.0	1612.7	17.6	18.70	18.70	19.0	19.0	0.0	1.6	137.1	3600.0
		$C3$	23.50	23.50	24.5	24.5	0.0	0.0	227.2	9.1	18.50	18.50	19.0	19.0	2.6	2.7	3600.0	3601.0
		$C4$	23.70	23.70	25.0	25.0	0.0	0.0	332.3	42.7	18.00	18.00	18.5	18.5	2.7	2.8	3600.5	3600.0
		$C5$	25.30	25.30	26.0	26.0	0.0	0.0	156.4	29.5	18.90	18.90	19.0	19.0	0.0	0.0	414.9	911.6
	$Q10$	$C1$	10.60	16.96	12.5	12.5	0.0	0.0	22.9	241.9	8.30	8.30	9.0	9.0	0.0	0.0	48.1	85.5
		$C2$	11.25	11.25	12.0	12.0	0.0	0.0	3.2	15.2	9.35	9.35	10.0	10.0	0.0	0.0	115.5	144.7
		$C3$	11.75	11.75	13.5	13.5	0.0	0.0	14.1	31.0	9.25	9.25	10.0	10.0	0.0	0.0	45.6	197.2
		$C4$	11.85	11.85	13.5	13.5	0.0	0.0	19.4	100.0	9.00	9.00	10.0	10.0	0.0	0.0	27.0	315.8
		$C5$	12.65	12.65	14.0	14.0	0.0	0.0	6.9	31.1	9.45	9.45	10.5	10.5	0.0	0.0	96.1	523.7
	$Q50$	$C1$	2.12	3.64	5.0	5.0	0.0	0.0	0.2	2.1	1.66	3.70	7.5	7.5	0.0	0.0	3.3	124.3
		$C2$	2.25	3.64	5.0	5.0	0.0	0.0	0.4	2.0	1.87	3.70	7.5	7.5	0.0	0.0	3.9	120.4
		$C3$	2.35	3.64	5.0	5.0	0.0	0.0	0.2	1.2	1.85	3.70	7.5	7.5	0.0	0.0	3.5	147.9
		$C4$	2.37	3.64	5.0	5.0	0.0	0.0	0.1	1.7	1.80	3.70	7.5	7.5	0.0	0.0	3.9	219.9
		$C5$	2.53	3.64	5.0	5.0	0.0	0.0	0.3	1.3	1.89	3.70	7.5	7.5	0.0	0.0	3.6	84.8

Table C.2.2: Branch-and-price results comparison for formulation MP^c ($\beta = 0.5$).

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP
$I10$	$Q5$	$C1$	4.80	4.80	4.8	4.8	0.0	0.0	0.0	0.0	4.00	4.06	5.2	5.2	0.0	0.0	0.7	5.2
		$C2$	6.08	6.31	8.4	8.4	0.0	0.0	0.9	8.3	5.60	5.88	9.0	9.0	0.0	0.0	3.7	90.2
		$C3$	5.44	5.44	6.6	6.6	0.0	0.0	0.3	1.7	5.92	6.12	9.2	9.2	0.0	0.0	1.4	67.5
		$C4$	6.88	7.02	9.0	9.0	0.0	0.0	1.8	4.5	5.76	6.09	9.4	9.4	0.0	0.0	3.4	114.9
		$C5$	8.16	8.22	9.2	9.2	0.0	0.0	0.8	1.2	6.24	6.24	8.0	8.0	0.0	0.0	1.5	7.0
	$Q10$	$C1$	2.40	4.00	4.0	4.0	0.0	0.0	0.2	0.4	2.00	2.60	4.4	4.4	0.0	0.0	0.8	4.7
		$C2$	3.04	5.83	7.6	7.6	0.0	0.0	0.6	17.1	2.80	5.87	9.0	9.0	0.0	0.0	2.3	266.9
		$C3$	2.72	3.80	3.8	3.8	0.0	0.0	0.2	0.2	2.96	5.92	8.8	8.8	0.0	0.0	0.8	118.6
		$C4$	3.44	5.83	7.8	7.8	0.0	0.0	0.6	9.6	2.88	5.92	9.4	9.4	0.0	0.0	1.5	377.7
		$C5$	4.08	5.83	8.0	8.0	0.0	0.0	1.2	3.9	3.12	4.80	7.6	7.6	0.0	0.0	0.6	29.8
$I30$	$Q5$	$C1$	17.12	17.12	18.4	18.4	0.0	0.0	371.2	28.2	13.44	13.44	14.4	14.4	0.0	0.0	30.0	1892.5
		$C2$	21.12	21.12	22.8	22.8	0.0	0.0	136.4	78.1	16.32	16.32	17.0	17.0	0.0	0.0	357.2	1523.4
		$C3$	20.48	20.48	21.6	21.6	0.0	0.0	14.1	88.4	16.96	16.96	17.6	17.6	0.0	0.0	85.8	2404.2
		$C4$	21.60	21.60	23.2	23.2	0.0	0.0	82.1	32.5	18.56	18.56	19.2	19.2	0.0	0.0	7.3	415.6
		$C5$	21.92	21.92	23.4	23.4	0.0	0.0	77.4	215.4	17.12	17.12	18.0	18.0	0.0	0.0	71.6	599.9
	$Q10$	$C1$	8.56	8.56	11.2	11.2	0.0	0.0	4.9	30.7	6.72	6.80	9.2	9.2	0.0	0.0	8.2	248.7
		$C2$	10.56	10.56	13.2	13.2	0.0	0.0	4.9	34.6	8.16	8.18	11.2	11.2	0.0	0.0	18.5	473.2
		$C3$	10.24	10.27	12.0	12.0	0.0	0.0	2.1	11.9	8.48	8.48	10.8	10.8	0.0	0.0	5.9	144.9
		$C4$	10.80	10.80	13.6	13.6	0.0	0.0	8.6	94.8	9.28	9.28	11.2	11.2	0.0	0.0	2.3	34.4
		$C5$	10.96	11.00	12.0	12.0	0.0	0.0	0.6	4.0	8.56	8.63	11.4	11.4	0.0	0.0	8.6	184.0
	$Q30$	$C1$	2.85	5.83	8.0	8.0	0.0	0.0	0.3	16.5	2.24	4.80	8.8	8.8	0.0	0.0	1.5	1306.7
		$C2$	3.52	5.83	8.0	8.0	0.0	0.0	0.2	3.3	2.72	5.92	11.2	11.2	0.0	0.0	1.6	2099.4
		$C3$	3.41	5.83	8.0	8.0	0.0	0.0	0.1	5.3	2.83	5.87	10.2	10.2	0.0	0.0	1.1	371.5
		$C4$	3.60	5.83	8.0	8.0	0.0	0.0	0.2	4.2	3.09	5.92	11.2	11.2	0.0	0.0	1.2	253.3
		$C5$	3.65	5.83	8.0	8.0	0.0	0.0	0.4	4.1	2.85	5.92	11.2	11.2	0.0	0.0	1.8	1020.3
$I50$	$Q5$	$C1$	33.92	33.92	35.0	35.0	0.0	0.0	2620.8	6.2	26.56	26.56	27.2	27.2	0.0	2.4	41.9	3600.0
		$C2$	36.00	36.00	37.2	37.2	1.2	0.0	3600.3	25.1	29.92	29.92	30.4	30.4	0.0	1.6	137.1	3600.0
		$C3$	37.60	37.60	38.8	38.8	0.0	0.0	1276.9	53.3	29.60	29.60	29.6	29.6	2.6	0.0	3600.0	7.5
		$C4$	37.92	37.92	39.2	39.2	0.0	0.0	1218.9	46.4	28.80	28.80	28.8	28.8	2.7	0.0	3600.5	5.2
		$C5$	40.48	40.48	41.0	41.0	0.0	0.0	210.6	6.5	30.24	30.24	30.4	30.4	0.0	0.0	414.9	3097.4
	$Q10$	$C1$	16.96	16.96	19.4	19.4	0.0	0.0	50.6	446.5	13.28	13.28	14.4	14.4	0.0	0.0	48.1	341.6
		$C2$	18.00	18.00	19.2	19.2	0.0	0.0	10.4	77.4	14.96	14.96	16.0	16.0	0.0	0.0	115.5	726.2
		$C3$	18.80	18.80	20.6	20.6	0.0	0.0	11.3	27.0	14.80	14.80	16.0	16.0	0.0	0.0	45.6	982.4
		$C4$	18.96	18.96	20.4	20.4	0.0	0.0	15.6	31.1	14.40	14.40	15.6	15.6	0.0	0.0	27.0	587.2
		$C5$	20.24	20.24	22.4	22.4	0.0	0.0	17.7	72.3	15.12	15.12	16.4	16.4	0.0	0.0	96.1	993.4
	$Q50$	$C1$	3.39	5.83	8.0	8.0	0.0	0.0	0.5	4.9	2.66	5.92	11.4	11.4	0.0	0.0	3.3	1228.8
		$C2$	3.60	5.83	8.0	8.0	0.0	0.0	0.4	3.3	2.99	5.92	12.0	12.0	0.0	0.0	3.9	921.0
		$C3$	3.76	5.83	8.0	8.0	0.0	0.0	0.3	3.0	2.96	5.92	11.6	11.6	0.0	0.0	3.5	587.2
		$C4$	3.79	5.83	8.0	8.0	0.0	0.0	0.4	2.1	2.88	5.92	11.8	11.8	0.0	0.0	3.9	1273.5
		$C5$	4.05	5.83	8.0	8.0	0.0	0.0	0.5	2.2	3.02	5.92	11.8	11.8	0.0	0.0	3.6	522.1

Table C.2.3: Branch-and-price results comparison for formulation MP^c ($\beta = 0.8$).

Instance			D-instances								U-instances							
			z_{LP}		UB		Gap		$Time$		z_{LP}		UB		Gap		$Time$	
			AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP	AS^c	MP
$I10$	$Q5$	$C1$	6.00	6.00	6.0	6.0	0.0	0.0	0.1	0.0	5.00	5.00	5.0	5.0	0.0	0.0	1.0	0.3
		$C2$	7.60	8.00	8.0	8.0	0.0	0.0	0.6	1.4	7.00	8.00	8.0	8.0	0.0	0.0	11.7	4.9
		$C3$	6.80	6.80	8.0	8.0	0.0	0.0	0.5	1.1	7.40	7.61	9.0	9.0	0.0	0.0	11.8	19.7
		$C4$	8.60	8.60	10.0	10.0	0.0	0.0	1.9	3.6	7.20	7.55	9.0	9.0	0.0	0.0	9.1	21.4
		$C5$	10.20	11.00	11.0	11.0	0.0	0.0	0.9	0.0	7.80	8.00	8.0	8.0	0.0	0.0	0.5	0.5
	$Q10$	$C1$	3.00	4.00	4.0	4.0	0.0	0.0	0.3	0.1	2.50	3.00	3.0	3.0	0.0	0.0	0.3	0.2
		$C2$	3.80	7.00	7.0	7.0	0.0	0.0	0.4	0.9	3.50	7.00	7.0	7.0	0.0	0.0	0.3	2.3
		$C3$	3.40	4.00	4.0	4.0	0.0	0.0	0.1	0.1	3.70	7.00	7.0	7.0	0.0	0.0	0.8	2.0
		$C4$	4.30	7.00	7.0	7.0	0.0	0.0	0.3	1.1	3.60	7.00	7.0	7.0	0.0	0.0	0.5	1.8
		$C5$	5.10	7.00	7.0	7.0	0.0	0.0	0.2	0.2	3.90	6.00	6.0	6.0	0.0	0.0	0.4	0.8
$I30$	$Q5$	$C1$	21.40	22.00	22.0	22.0	0.0	0.0	1.0	0.9	16.80	17.00	17.0	17.0	0.0	0.0	2.2	2.6
		$C2$	26.40	27.00	27.0	27.0	0.0	0.0	0.7	1.0	20.40	21.00	21.0	21.0	0.0	0.0	3.9	2.8
		$C3$	25.60	26.00	26.0	26.0	0.0	0.0	0.6	1.0	21.20	22.00	22.0	22.0	0.0	0.0	1.4	2.9
		$C4$	27.00	27.00	27.0	27.0	0.0	0.0	1.3	1.1	23.20	24.00	24.0	24.0	0.0	0.0	1.0	2.6
		$C5$	27.40	28.00	28.0	28.0	0.0	0.0	0.7	1.0	21.40	22.00	22.0	22.0	0.0	0.0	8.6	3.7
	$Q10$	$C1$	10.70	10.70	12.0	12.0	0.0	0.0	2.5	12.2	8.40	9.00	9.0	9.0	0.0	0.0	3.8	4.4
		$C2$	13.20	14.00	14.0	14.0	0.0	0.0	1.6	1.0	10.20	11.00	11.0	11.0	0.0	0.0	8.6	4.3
		$C3$	12.80	12.80	14.0	14.0	0.0	0.0	4.7	17.6	10.60	11.00	11.0	11.0	0.0	0.0	4.6	4.3
		$C4$	13.50	13.50	15.0	15.0	0.0	0.0	4.5	19.2	11.60	12.00	12.0	12.0	0.0	0.0	20.3	3.1
		$C5$	13.70	13.70	15.0	15.0	0.0	0.0	4.2	15.6	10.70	10.70	12.0	12.0	0.0	0.0	41.5	673.6
	$Q30$	$C1$	3.57	7.00	7.0	7.0	0.0	0.0	0.3	1.6	2.80	6.00	6.0	6.0	0.0	0.0	1.8	3.0
		$C2$	4.40	7.00	7.0	7.0	0.0	0.0	0.2	1.1	3.40	7.00	7.0	7.0	0.0	0.0	1.6	4.0
		$C3$	4.27	7.00	7.0	7.0	0.0	0.0	0.2	0.9	3.53	7.00	7.0	7.0	0.0	0.0	0.4	2.7
		$C4$	4.50	7.00	7.0	7.0	0.0	0.0	0.2	0.7	3.87	7.00	7.0	7.0	0.0	0.0	0.5	1.3
		$C5$	4.57	7.00	7.0	7.0	0.0	0.0	0.3	0.6	3.57	7.00	7.0	7.0	0.0	0.0	2.5	3.0
$I50$	$Q5$	$C1$	42.40	43.00	43.0	43.0	0.0	0.0	1.0	1.5	33.20	34.00	34.0	34.0	0.0	0.0	6.8	4.2
		$C2$	45.00	45.00	45.0	45.0	0.0	0.0	1.2	1.0	37.40	38.00	38.0	38.0	0.0	0.0	3.5	5.9
		$C3$	47.00	47.00	47.0	47.0	0.0	0.0	3.0	1.0	37.00	37.00	37.0	37.0	0.0	0.0	11.1	2.8
		$C4$	47.40	48.00	48.0	48.0	0.0	0.0	1.8	1.3	36.00	36.00	36.0	36.0	0.0	0.0	15.7	3.2
		$C5$	50.60	51.00	51.0	51.0	0.0	0.0	0.2	1.5	37.80	38.00	38.0	38.0	0.0	0.0	14.6	3.6
	$Q10$	$C1$	21.20	22.00	22.0	22.0	0.0	0.0	1.0	1.0	16.60	17.00	17.0	17.0	0.0	0.0	28.6	3.6
		$C2$	22.50	23.00	23.0	23.0	0.0	0.0	0.6	0.9	18.70	19.00	19.0	19.0	0.0	0.0	13.4	3.7
		$C3$	23.50	24.00	24.0	24.0	0.0	0.0	1.1	1.5	18.50	19.00	19.0	19.0	0.0	0.0	4.0	3.0
		$C4$	23.70	24.00	24.0	24.0	0.0	0.0	1.1	1.1	18.00	18.00	18.0	18.0	0.0	0.0	17.8	2.7
		$C5$	25.30	26.00	26.0	26.0	0.0	0.0	0.8	0.9	18.90	19.00	19.0	19.0	0.0	0.0	17.6	4.1
	$Q50$	$C1$	4.24	7.00	7.0	7.0	0.0	0.0	0.2	1.1	3.32	7.00	7.0	7.0	0.0	0.0	2.0	2.4
		$C2$	4.50	7.00	7.0	7.0	0.0	0.0	0.2	0.5	3.74	7.00	7.0	7.0	0.0	0.0	1.9	3.4
		$C3$	4.70	7.00	7.0	7.0	0.0	0.0	0.3	0.5	3.70	7.00	7.0	7.0	0.0	0.0	1.1	2.0
		$C4$	4.74	7.00	7.0	7.0	0.0	0.0	0.2	0.6	3.60	7.00	7.0	7.0	0.0	0.0	1.5	3.0
		$C5$	5.06	7.00	7.0	7.0	0.0	0.0	0.3	0.5	3.78	7.00	7.0	7.0	0.0	0.0	2.0	1.6

Table C.2.4: Branch-and-price results comparison for formulation MP^c ($\beta = 1.0$).

Appendix D

Heuristic Algorithm Results

D.1 Detailed relative contribution of each heuristic phase

	Instance	PS policy			AS policy											
					$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$		
		HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P
I10	D_I10_Q5_C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	D_I10_Q5_C2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.1	4.8	0.0	25.0	12.5	0.0
	D_I10_Q5_C3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.1	0.0	0.0	0.0	0.0	0.0
	D_I10_Q5_C4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.1	0.0	0.0	0.0	0.0	0.0
	D_I10_Q5_C5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	D_I10_Q10_C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0
	D_I10_Q10_C2	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	15.8	0.0	0.0	0.0	0.0	0.0
	D_I10_Q10_C3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	D_I10_Q10_C4	10.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	10.3	0.0	0.0	0.0	0.0	0.0
	D_I10_Q10_C5	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	2.5	0.0	0.0	0.0	0.0	0.0
I30	D_I30_Q5_C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.7	0.0	0.0	4.5	4.5	0.0
	D_I30_Q5_C2	6.9	0.0	0.0	3.4	0.0	0.0	6.9	0.0	0.0	21.9	0.9	0.9	3.7	3.7	0.0
	D_I30_Q5_C3	3.6	0.0	0.0	3.6	0.0	0.0	3.6	0.0	0.0	14.8	2.8	2.8	3.8	3.8	0.0
	D_I30_Q5_C4	6.7	3.3	0.0	6.7	3.3	0.0	6.7	0.0	0.0	23.3	4.3	4.3	11.1	7.4	0.0
	D_I30_Q5_C5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.6	0.0	0.0	0.0	0.0	0.0
	D_I30_Q10_C1	14.3	7.1	0.0	7.1	0.0	0.0	14.3	0.0	0.0	50.0	3.6	3.6	8.3	0.0	0.0
	D_I30_Q10_C2	11.8	5.9	0.0	5.9	0.0	0.0	11.8	5.9	0.0	51.5	3.0	3.0	7.1	7.1	0.0
	D_I30_Q10_C3	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	16.7	0.0	0.0	0.0	0.0	0.0
	D_I30_Q10_C4	5.6	0.0	0.0	5.6	0.0	0.0	5.6	0.0	0.0	60.3	0.0	0.0	13.3	0.0	0.0
	D_I30_Q10_C5	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	10.0	5.0	5.0	0.0	0.0	0.0
	D_I30_Q30_C1	10.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	52.5	0.0	0.0	0.0	0.0	0.0
	D_I30_Q30_C2	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	65.0	0.0	0.0	0.0	0.0	0.0
	D_I30_Q30_C3	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	45.0	0.0	0.0	0.0	0.0	0.0
	D_I30_Q30_C4	10.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	37.5	25.0	0.0	0.0	0.0	0.0
	D_I30_Q30_C5	10.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	42.5	35.0	0.0	0.0	0.0	0.0
I50	D_I50_Q5_C1	2.3	2.3	0.0	2.3	2.3	0.0	2.3	2.3	0.0	2.3	0.6	0.6	0.0	0.0	0.0
	D_I50_Q5_C2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.8	0.5	0.0	2.2	2.2	0.0
	D_I50_Q5_C3	2.0	0.0	0.0	2.0	0.0	0.0	2.0	0.0	0.0	8.2	0.5	0.0	4.3	4.3	0.0
	D_I50_Q5_C4	8.0	4.0	0.0	4.0	0.0	0.0	8.0	0.0	0.0	13.3	0.0	0.0	4.2	4.2	0.0
	D_I50_Q5_C5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0
	D_I50_Q10_C1	8.0	4.0	0.0	4.0	0.0	0.0	8.0	0.0	0.0	56.7	2.1	2.1	4.5	4.5	0.0
	D_I50_Q10_C2	4.2	0.0	0.0	4.2	0.0	0.0	4.2	0.0	0.0	19.8	0.0	0.0	4.3	4.3	0.0
	D_I50_Q10_C3	3.7	0.0	0.0	3.7	0.0	0.0	3.7	0.0	0.0	46.6	3.9	3.9	8.3	8.3	0.0
	D_I50_Q10_C4	7.1	3.6	0.0	3.6	0.0	0.0	11.1	7.4	0.0	53.9	0.0	0.0	8.3	8.3	0.0
	D_I50_Q10_C5	7.1	3.6	0.0	3.6	0.0	0.0	7.1	3.6	0.0	19.6	4.5	4.5	3.8	3.8	0.0
	U_I50_Q50_C1	10.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	117.5	10.0	0.0	0.0	0.0	0.0
	U_I50_Q50_C2	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	67.5	37.5	0.0	0.0	0.0	0.0
	U_I50_Q50_C3	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	57.5	25.0	0.0	0.0	0.0	0.0
	U_I50_Q50_C4	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	62.5	40.0	0.0	0.0	0.0	0.0
	U_I50_Q50_C5	10.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	52.5	35.0	0.0	0.0	0.0	0.0

Table D.1.1: Relative contribution of each heuristic phase (D-instances).

	Instance	PS policy			AS policy											
					$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$		
		HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P	HG_G	HG_{LS}	HG_P
$I10$	U_I10_Q5_C1	14.3	0.0	0.0	14.3	0.0	0.0	14.3	0.0	0.0	34.6	0.0	0.0	0.0	0.0	0.0
	U_I10_Q5_C2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.2	0.0	0.0	12.5	12.5	0.0
	U_I10_Q5_C3	8.3	0.0	0.0	0.0	0.0	0.0	8.3	0.0	0.0	23.9	0.0	0.0	0.0	0.0	0.0
	U_I10_Q5_C4	15.4	0.0	0.0	15.4	0.0	0.0	15.4	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0
	U_I10_Q5_C5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27.5	0.0	0.0	0.0	0.0	0.0
	U_I10_Q10_C1	14.3	0.0	0.0	14.3	0.0	0.0	14.3	0.0	0.0	40.9	0.0	0.0	0.0	0.0	0.0
	U_I10_Q10_C2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	U_I10_Q10_C3	8.3	0.0	0.0	0.0	0.0	0.0	8.3	0.0	0.0	6.8	0.0	0.0	0.0	0.0	0.0
	U_I10_Q10_C4	15.4	0.0	0.0	15.4	0.0	0.0	15.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	U_I10_Q10_C5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	13.2	0.0	0.0	0.0	0.0	0.0
$I30$	U_I30_Q5_C1	11.1	5.6	0.0	5.6	5.6	0.0	11.1	5.6	0.0	27.8	2.8	2.8	5.9	5.9	0.0
	U_I30_Q5_C2	9.1	0.0	0.0	9.1	0.0	0.0	9.1	0.0	0.0	68.2	4.7	4.7	4.8	4.8	0.0
	U_I30_Q5_C3	9.1	4.5	0.0	9.1	4.5	0.0	9.1	9.1	0.0	23.9	2.3	2.3	4.5	0.0	0.0
	U_I30_Q5_C4	4.2	0.0	0.0	8.3	0.0	0.0	8.3	4.2	0.0	24.0	0.0	0.0	0.0	0.0	0.0
	U_I30_Q5_C5	4.3	4.3	0.0	13.0	4.3	0.0	4.3	4.3	0.0	50.0	5.6	5.6	4.5	4.5	0.0
	U_I30_Q10_C1	16.7	8.3	0.0	8.3	8.3	0.0	16.7	8.3	0.0	41.3	15.2	0.0	22.2	0.0	0.0
	U_I30_Q10_C2	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	75.0	7.1	0.0	9.1	9.1	0.0
	U_I30_Q10_C3	7.1	0.0	0.0	7.1	0.0	0.0	7.1	7.1	0.0	38.9	1.9	1.9	18.2	9.1	0.0
	U_I30_Q10_C4	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	76.8	8.9	5.4	16.7	8.3	0.0
	U_I30_Q10_C5	13.3	0.0	0.0	13.3	0.0	0.0	13.3	6.7	0.0	82.5	5.3	5.3	8.3	0.0	0.0
	U_I30_Q30_C1	8.3	0.0	0.0	8.3	0.0	0.0	8.3	0.0	0.0	22.7	11.4	0.0	0.0	0.0	0.0
	U_I30_Q30_C2	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	57.1	14.3	0.0	0.0	0.0	0.0
	U_I30_Q30_C3	7.7	0.0	0.0	7.7	0.0	0.0	7.7	0.0	0.0	33.3	5.9	0.0	0.0	0.0	0.0
	U_I30_Q30_C4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	44.6	10.7	5.4	0.0	0.0	0.0
	U_I30_Q30_C5	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	30.4	8.9	5.4	0.0	0.0	0.0
$I50$	U_I50_Q5_C1	5.9	0.0	0.0	11.8	0.0	0.0	5.9	5.9	0.0	36.8	0.0	0.0	2.9	2.9	0.0
	U_I50_Q5_C2	7.9	2.6	0.0	7.9	2.6	0.0	7.9	5.3	0.0	27.0	3.3	3.3	0.0	0.0	0.0
	U_I50_Q5_C3	10.8	5.4	2.7	5.3	2.6	0.0	10.5	5.3	0.0	34.5	2.7	0.0	2.7	2.7	0.0
	U_I50_Q5_C4	5.6	5.6	2.8	8.1	2.7	0.0	8.1	2.7	0.0	29.9	2.8	0.0	5.6	5.6	0.0
	U_I50_Q5_C5	7.9	2.6	0.0	7.9	2.6	0.0	7.9	7.9	0.0	41.4	3.3	0.0	2.6	2.6	0.0
	U_I50_Q10_C1	16.7	5.6	0.0	11.1	11.1	0.0	16.7	11.1	0.0	81.9	5.6	0.0	11.8	5.9	0.0
	U_I50_Q10_C2	10.0	5.0	0.0	15.0	10.0	0.0	10.0	10.0	0.0	153.8	5.0	5.0	10.5	5.3	0.0
	U_I50_Q10_C3	15.0	5.0	0.0	15.0	5.0	0.0	15.0	5.0	0.0	85.0	3.8	3.8	0.0	0.0	0.0
	U_I50_Q10_C4	15.0	10.0	0.0	10.0	0.0	0.0	15.0	15.0	0.0	141.0	2.6	2.6	16.7	11.1	0.0
	U_I50_Q10_C5	9.5	4.8	0.0	4.8	0.0	0.0	9.5	9.5	0.0	85.4	4.9	0.0	5.3	5.3	0.0
	U_I50_Q50_C1	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	50.9	1.8	1.8	0.0	0.0	0.0
	U_I50_Q50_C2	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	120.0	11.7	5.0	0.0	0.0	0.0
	U_I50_Q50_C3	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	60.3	8.6	0.0	0.0	0.0	0.0
	U_I50_Q50_C4	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	96.6	20.3	0.0	0.0	0.0	0.0
	U_I50_Q50_C5	6.7	0.0	0.0	6.7	0.0	0.0	6.7	0.0	0.0	88.1	13.6	5.1	0.0	0.0	0.0

Table D.1.2: Relative contribution of each heuristic phase (U-instances).

D.2 Detailed results for the heuristic algorithm

Instance		PS policy		AS policy													
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$		
		z	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$
I10	D_I10_Q5_C1	6	2.6	6	0	2.3	6	0	2.0	6	0	1.7	6	0	2.0
	D_I10_Q5_C2	12	1.4	12	0	3.3	12	0	3.6	10	2	2.6	8	17	2.3
	D_I10_Q5_C3	9	2.5	9	0	2.2	9	0	2.2	8	1	2.3	8	3	1.6
	D_I10_Q5_C4	12	1.4	12	0	4.0	12	0	4.1	11	1	3.6	10	47	2.4
	D_I10_Q5_C5	12	1.8	12	0	3.2	12	0	3.0	11	2	2.4	11	2	2.4
	D_I10_Q10_C1	5	3.5	5	0	1.9	5	0	1.8	5	0	2.0	4	5	2.1
	D_I10_Q10_C2	10	2.1	10	0	3.2	10	0	3.5	9	2	2.8	7	72	2.7
	D_I10_Q10_C3	5	3.6	5	0	2.0	5	0	2.0	4	3	1.5	4	3	1.5
	D_I10_Q10_C4	10	2.0	10	0	3.7	10	0	3.9	9	3	12.8	7	69	2.5
	D_I10_Q10_C5	10	3.6	10	0	3.7	10	0	4.0	10	0	5.1	7	51	2.4
I30	D_I30_Q5_C1	23	12.3	23	0	22.9	23	0	31.7	23	0	12.9	22	66	31.1
	D_I30_Q5_C2	29	11.0	29	0	54.8	29	0	180.8	28	3	518.0	27	79	26.4
	D_I30_Q5_C3	28	6.9	28	0	20.4	28	0	41.7	27	3	487.0	26	43	25.8
	D_I30_Q5_C4	30	8.4	30	0	22.8	30	0	38.2	30	1	66.3	27	65	27.6
	D_I30_Q5_C5	30	7.2	30	0	22.4	30	0	39.8	29	1	23.4	28	17	26.5
	D_I30_Q10_C1	14	12.5	14	0	24.7	14	0	39.7	14	2	35.6	12	144	64.8
	D_I30_Q10_C2	17	24.7	17	0	18.3	17	0	51.2	17	0	88.3	14	51	19.8
	D_I30_Q10_C3	15	11.4	15	0	19.3	15	0	33.9	15	0	16.1	14	66	23.1
	D_I30_Q10_C4	18	17.2	18	0	22.5	18	0	38.7	17	0	51.1	15	56	20.5
	D_I30_Q10_C5	15	12.7	15	0	18.9	15	0	37.6	15	3	27.3	15	20	22.5
	D_I30_Q30_C1	10	16.6	10	0	27.6	10	0	29.7	10	0	29.8	7	207	56.8
	D_I30_Q30_C2	10	16.5	10	0	19.0	10	0	29.5	10	0	41.1	7	126	21.6
	D_I30_Q30_C3	10	9.8	10	0	21.0	10	0	29.3	10	0	30.7	7	234	21.6
	D_I30_Q30_C4	10	11.4	10	0	26.2	10	0	29.7	10	0	43.4	7	177	21.2
	D_I30_Q30_C5	10	14.3	10	0	27.7	10	0	27.2	10	0	37.4	7	255	20.9
I50	D_I50_Q5_C1	44	17.6	44	0	12.2	44	0	47.3	44	0	40.3	43	95	41.0
	D_I50_Q5_C2	47	17.9	47	0	13.8	47	0	18.4	46	2	41.7	45	135	50.8
	D_I50_Q5_C3	49	52.8	49	0	31.5	49	0	57.5	48	2	69.6	47	117	50.8
	D_I50_Q5_C4	50	24.0	50	0	56.6	50	0	95.9	49	0	77.9	48	104	39.1
	D_I50_Q5_C5	52	39.1	52	0	27.3	52	0	46.9	51	1	61.3	51	2	43.9
	D_I50_Q10_C1	25	57.1	25	0	47.1	25	0	65.6	24	3	20.3	22	138	37.1
	D_I50_Q10_C2	24	72.5	24	0	64.7	24	0	64.3	24	0	68.2	23	123	43.6
	D_I50_Q10_C3	27	58.4	27	0	58.5	27	0	54.9	26	3	139.2	24	100	62.9
	D_I50_Q10_C4	28	41.8	28	0	66.3	27	0	47.7	25	2	52.2	24	17	44.5
	D_I50_Q10_C5	28	52.4	28	0	48.7	28	0	50.4	29	1	165.5	26	107	34.2
	U_I50_Q50_C1	10	60.4	10	0	29.1	10	0	68.4	10	0	58.0	7	360	72.4
	U_I50_Q50_C2	10	76.6	10	0	20.8	10	0	83.0	10	0	81.4	7	273	70.1
	U_I50_Q50_C3	10	59.5	10	0	29.8	10	0	64.1	10	0	67.7	7	201	63.7
	U_I50_Q50_C4	10	43.5	10	0	19.9	10	0	68.3	10	0	75.8	7	279	52.7
	U_I50_Q50_C5	10	53.8	10	0	19.0	10	0	80.9	10	0	46.7	7	243	54.5

Table D.2.1: Heuristic results for the P-MSSP and A-MSSP (D-instances).

	Instance	PS policy		AS policy											
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$		
		z	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$	z	e	$Time$
I10	U_I10_Q5_C1	7	1.6	7	0	3.0	7	0	3.4	6	2	1.9	5	15	1.7
	U_I10_Q5_C2	13	1.6	13	0	3.6	13	0	3.6	11	1	2.6	8	52	2.2
	U_I10_Q5_C3	12	2.5	12	0	3.3	12	0	3.5	11	2	4.2	9	28	2.2
	U_I10_Q5_C4	13	1.6	13	0	3.4	13	0	3.9	11	3	3.1	9	25	2.2
	U_I10_Q5_C5	10	1.9	10	0	2.8	10	0	3.2	10	0	3.0	8	19	2.1
	U_I10_Q10_C1	7	4.6	7	0	2.8	7	0	2.8	5	2	2.5	3	27	1.7
	U_I10_Q10_C2	13	2.5	13	0	4.1	13	0	4.2	11	1	3.3	7	51	2.8
	U_I10_Q10_C3	12	4.1	12	0	3.9	12	0	4.2	11	0	4.0	7	34	2.7
	U_I10_Q10_C4	13	2.4	13	0	4.0	13	0	4.5	11	3	3.4	7	35	2.7
	U_I10_Q10_C5	10	5.3	10	0	3.1	10	0	3.7	9	2	15.9	6	22	2.4
I30	U_I30_Q5_C1	18	7.4	18	0	16.8	18	0	47.6	18	2	33.7	17	59	25.4
	U_I30_Q5_C2	22	8.1	22	0	94.1	22	0	107.8	22	1	129.0	21	83	28.9
	U_I30_Q5_C3	22	9.6	22	0	21.1	22	0	44.3	22	2	54.0	22	76	56.5
	U_I30_Q5_C4	24	17.4	24	0	22.3	24	0	44.6	24	0	29.9	24	23	33.5
	U_I30_Q5_C5	23	5.1	23	0	84.0	23	0	47.0	23	3	46.9	22	81	36.3
	U_I30_Q10_C1	12	16.8	12	0	24.8	12	0	57.2	11	2	55.9	9	69	51.6
	U_I30_Q10_C2	15	49.0	15	0	15.8	15	0	61.8	14	0	32.7	11	99	20.7
	U_I30_Q10_C3	14	22.4	14	0	21.2	14	0	37.0	13	3	29.2	11	65	23.1
	U_I30_Q10_C4	15	75.4	15	0	19.5	15	0	124.7	14	3	22.5	12	67	27.5
	U_I30_Q10_C5	15	48.8	15	0	19.8	15	0	51.9	15	0	29.4	12	84	26.1
	U_I30_Q30_C1	12	17.5	12	0	36.1	12	0	96.6	11	0	76.4	6	173	28.6
	U_I30_Q30_C2	15	22.7	15	0	17.0	15	0	43.7	14	0	31.0	7	157	37.0
	U_I30_Q30_C3	13	24.1	13	0	31.0	13	0	54.3	12	3	45.5	7	167	65.4
	U_I30_Q30_C4	15	17.1	15	0	17.0	15	0	48.9	14	3	27.8	7	145	54.8
	U_I30_Q30_C5	15	52.8	15	0	16.2	15	0	40.2	14	3	33.2	7	129	59.1
I50	U_I50_Q5_C1	34	51.8	34	0	201.4	34	0	70.0	34	0	10.7	34	103	74.9
	U_I50_Q5_C2	38	75.8	38	0	70.0	38	0	71.6	39	1	8.4	38	41	44.0
	U_I50_Q5_C3	38	52.8	38	0	84.4	38	0	121.0	37	0	8.5	37	156	51.1
	U_I50_Q5_C4	37	24.0	37	0	104.1	37	0	89.6	36	0	6.9	36	78	40.5
	U_I50_Q5_C5	38	39.1	38	0	83.1	38	0	61.7	38	0	12.7	38	151	47.4
	U_I50_Q10_C1	18	57.6	18	0	45.9	18	0	54.0	18	0	168.7	17	63	72.1
	U_I50_Q10_C2	20	95.3	20	0	86.5	20	0	58.4	21	0	184.2	19	123	57.3
	U_I50_Q10_C3	20	70.4	20	0	86.7	20	0	108.7	20	3	135.1	19	72	67.5
	U_I50_Q10_C4	20	58.5	20	0	47.0	20	0	55.3	20	0	154.4	18	139	42.4
	U_I50_Q10_C5	21	96.4	21	0	75.6	21	0	51.9	20	2	118.4	19	93	71.4
	U_I50_Q50_C1	15	61.9	15	0	39.7	15	0	162.8	14	2	138.1	7	159	101.3
	U_I50_Q50_C2	15	64.3	15	0	41.4	15	0	100.8	15	3	155.8	7	252	71.1
	U_I50_Q50_C3	15	47.4	15	0	42.7	15	0	230.4	14	2	293.5	7	234	136.5
	U_I50_Q50_C4	15	40.1	15	0	42.2	15	0	349.8	14	3	190.0	7	212	196.0
	U_I50_Q50_C5	15	67.6	15	0	42.1	15	0	253.1	15	2	318.5	7	197	147.6

Table D.2.2: Heuristic results for the P-MSSP and A-MSSP (U-instances).

D.3 Heuristic algorithm comparison for the P-MSSP

Instance		Sparse formulations				Dense formulations			
		PS		PS^c		PD		PD^c	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
$I10$	$D\_I10\_Q5\_C1$	0.0	2.1	0.0	2.5	0.0	2.5	0.0	2.5
	$D\_I10\_Q5\_C2$	0.0	0.4	0.0	1.1	0.0	1.2	0.0	1.3
	$D\_I10\_Q5\_C3$	0.0	1.4	0.0	2.3	0.0	2.4	0.0	2.4
	$D\_I10\_Q5\_C4$	0.0	0.6	0.0	1.2	0.0	1.4	0.0	1.3
	$D\_I10\_Q5\_C5$	0.0	1.1	0.0	1.5	0.0	1.8	0.0	1.7
	$D\_I10\_Q10\_C1$	0.0	3.3	0.0	3.4	0.0	3.2	0.0	3.4
	$D\_I10\_Q10\_C2$	0.0	2.0	0.0	1.8	0.0	2.0	0.0	1.8
	$D\_I10\_Q10\_C3$	0.0	3.3	0.0	3.5	0.0	3.3	0.0	3.5
	$D\_I10\_Q10\_C4$	0.0	1.9	0.0	1.9	0.0	1.8	0.0	1.9
	$D\_I10\_Q10\_C5$	0.0	3.4	0.0	3.4	0.0	3.4	0.0	3.5
$I30$	$D\_I30\_Q5\_C1$	0.0	-804.9	0.0	3.7	0.0	10.2	0.0	11.2
	$D\_I30\_Q5\_C2$	0.0	-1539.4	0.0	2.4	0.0	9.2	0.0	10.1
	$D\_I30\_Q5\_C3$	0.0	-1165.0	0.0	-1.3	0.0	5.6	0.0	5.9
	$D\_I30\_Q5\_C4$	0.0	-2219.5	0.0	-5.2	0.0	1.2	0.0	5.8
	$D\_I30\_Q5\_C5$	0.0	-2063.6	0.0	-9.9	0.0	3.9	0.0	5.6
	$D\_I30\_Q10\_C1$	0.0	8.0	0.0	11.9	0.0	12.2	0.0	12.2
	$D\_I30\_Q10\_C2$	0.0	-11.0	0.0	24.1	0.0	24.4	0.0	24.6
	$D\_I30\_Q10\_C3$	0.0	5.6	0.0	11.2	0.0	11.3	0.0	11.4
	$D\_I30\_Q10\_C4$	0.0	-30.0	0.0	16.1	0.0	16.8	0.0	17.1
	$D\_I30\_Q10\_C5$	0.0	10.1	0.0	12.5	0.0	12.6	0.0	12.6
	$D\_I30\_Q30\_C1$	0.0	16.3	0.0	16.4	0.0	16.4	0.0	16.3
	$D\_I30\_Q30\_C2$	0.0	16.3	0.0	16.3	0.0	16.3	0.0	16.3
	$D\_I30\_Q30\_C3$	0.0	9.7	0.0	9.7	0.0	9.7	0.0	9.8
	$D\_I30\_Q30\_C4$	0.0	11.3	0.0	11.3	0.0	11.3	0.0	11.4
	$D\_I30\_Q30\_C5$	0.0	14.3	0.0	14.3	0.0	14.2	0.0	14.3
$I50$	$D\_I50\_Q5\_C1$	-2.3	-3584.1	0.0	-339.1	0.0	-906.1	0.0	-26.4
	$D\_I50\_Q5\_C2$	0.0	-3584.4	0.0	-306.2	0.0	-834.6	0.0	2.8
	$D\_I50\_Q5\_C3$	0.0	-3549.3	0.0	-83.6	0.0	40.4	0.0	49.1
	$D\_I50\_Q5\_C4$	0.0	-3577.9	0.0	-72.8	0.0	-0.1	0.0	15.3
	$D\_I50\_Q5\_C5$	0.0	-3562.7	0.0	-1.1	0.0	15.8	0.0	35.6
	$D\_I50\_Q10\_C1$	0.0	-1547.4	0.0	49.1	0.0	52.0	0.0	55.4
	$D\_I50\_Q10\_C2$	0.0	-361.1	0.0	70.2	0.0	72.0	0.0	72.4
	$D\_I50\_Q10\_C3$	0.0	-989.6	0.0	54.8	0.0	56.5	0.0	58.0
	$D\_I50\_Q10\_C4$	0.0	-3559.1	0.0	33.9	0.0	39.5	0.0	40.6
	$D\_I50\_Q10\_C5$	0.0	-180.1	0.0	50.0	0.0	51.9	0.0	52.2
	$D\_I50\_Q50\_C1$	0.0	60.3	0.0	60.3	0.0	60.2	0.0	60.3
	$D\_I50\_Q50\_C2$	0.0	76.5	0.0	76.5	0.0	76.4	0.0	76.5
	$D\_I50\_Q50\_C3$	0.0	59.3	0.0	59.2	0.0	59.3	0.0	59.2
	$D\_I50\_Q50\_C4$	0.0	43.4	0.0	43.4	0.0	43.3	0.0	43.5
	$D\_I50\_Q50\_C5$	0.0	53.6	0.0	53.7	0.0	53.6	0.0	53.7

Table D.3.1: Heuristic comparison for the P-MSSP formulations (D-instances).

	Instance	Sparse formulations				Dense formulations			
		PS		PS^c		PD		PD^c	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
$I10$	$U\_I10\_Q5\_C1$	0.0	1.0	0.0	1.0	0.0	1.3	0.0	1.4
	$U\_I10\_Q5\_C2$	0.0	0.9	0.0	0.6	0.0	1.5	0.0	1.5
	$U\_I10\_Q5\_C3$	0.0	2.0	0.0	2.0	0.0	2.4	0.0	2.3
	$U\_I10\_Q5\_C4$	0.0	1.1	0.0	0.7	0.0	1.5	0.0	1.4
	$U\_I10\_Q5\_C5$	0.0	1.5	0.0	1.6	0.0	1.8	0.0	1.9
	$U\_I10\_Q10\_C1$	0.0	4.4	0.0	4.5	0.0	4.4	0.0	4.4
	$U\_I10\_Q10\_C2$	0.0	2.4	0.0	2.4	0.0	2.4	0.0	2.4
	$U\_I10\_Q10\_C3$	0.0	3.8	0.0	3.8	0.0	3.9	0.0	3.8
	$U\_I10\_Q10\_C4$	0.0	2.3	0.0	2.3	0.0	2.3	0.0	2.3
	$U\_I10\_Q10\_C5$	0.0	5.1	0.0	5.1	0.0	5.3	0.0	5.2
$I30$	$U\_I30\_Q5\_C1$	0.0	-103.6	0.0	-7.3	0.0	6.0	0.0	6.7
	$U\_I30\_Q5\_C2$	0.0	-847.5	0.0	-89.4	0.0	2.6	0.0	3.4
	$U\_I30\_Q5\_C3$	0.0	-112.4	0.0	-30.5	0.0	7.9	0.0	5.9
	$U\_I30\_Q5\_C4$	0.0	-111.0	0.0	11.6	0.0	16.9	0.0	17.3
	$U\_I30\_Q5\_C5$	0.0	-399.8	0.0	-41.6	0.0	2.6	0.0	3.7
	$U\_I30\_Q10\_C1$	0.0	10.8	0.0	13.5	0.0	16.7	0.0	16.7
	$U\_I30\_Q10\_C2$	0.0	30.9	0.0	47.7	0.0	48.7	0.0	48.8
	$U\_I30\_Q10\_C3$	0.0	11.0	0.0	21.1	0.0	22.2	0.0	22.1
	$U\_I30\_Q10\_C4$	0.0	64.6	0.0	74.2	0.0	75.1	0.0	75.1
	$U\_I30\_Q10\_C5$	0.0	38.7	0.0	46.8	0.0	48.5	0.0	48.3
	$U\_I30\_Q30\_C1$	0.0	17.2	0.0	17.2	0.0	17.0	0.0	17.2
	$U\_I30\_Q30\_C2$	0.0	22.5	0.0	22.6	0.0	22.3	0.0	22.5
	$U\_I30\_Q30\_C3$	0.0	23.9	0.0	23.9	0.0	23.7	0.0	24.0
	$U\_I30\_Q30\_C4$	0.0	16.9	0.0	16.8	0.0	16.9	0.0	16.8
	$U\_I30\_Q30\_C5$	0.0	52.5	0.0	52.7	0.0	52.6	0.0	52.7
$I50$	$U\_I50\_Q5\_C1$	0.0	-70.6	0.0	26.5	0.0	49.4	0.0	51.1
	$U\_I50\_Q5\_C2$	0.0	-775.5	0.0	51.2	0.0	73.9	0.0	74.6
	$U\_I50\_Q5\_C3$	0.0	-3547.3	0.0	-3548.0	2.6	-17.9	2.6	37.8
	$U\_I50\_Q5\_C4$	0.0	-3576.0	0.0	-3576.2	2.7	-5.7	2.7	-371.9
	$U\_I50\_Q5\_C5$	-2.6	-3562.9	0.0	-68.9	0.0	19.7	0.0	35.4
	$U\_I50\_Q10\_C1$	0.0	-101.5	0.0	43.6	0.0	56.9	0.0	56.8
	$U\_I50\_Q10\_C2$	0.0	-474.4	0.0	73.3	0.0	92.1	0.0	93.9
	$U\_I50\_Q10\_C3$	0.0	-155.8	0.0	50.1	0.0	69.5	0.0	69.5
	$U\_I50\_Q10\_C4$	0.0	-269.6	0.0	40.2	0.0	55.1	0.0	57.2
	$U\_I50\_Q10\_C5$	0.0	-434.3	0.0	36.6	0.0	92.1	0.0	94.7
	$U\_I50\_Q50\_C1$	0.0	61.5	0.0	61.6	0.0	61.5	0.0	61.7
	$U\_I50\_Q50\_C2$	0.0	64.0	0.0	64.2	0.0	63.8	0.0	64.2
	$U\_I50\_Q50\_C3$	0.0	47.0	0.0	47.1	0.0	47.0	0.0	47.3
	$U\_I50\_Q50\_C4$	0.0	39.6	0.0	39.9	0.0	39.8	0.0	39.9
	$U\_I50\_Q50\_C5$	0.0	67.1	0.0	67.5	0.0	67.1	0.0	67.5

Table D.3.2: Heuristic comparison for the P-MSSP formulations (U-instances).

D.4 Heuristic algorithm comparison for the A-MSSP

	Instance	AS								AS^c							
		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
$I10$	D_I10_Q5_C1	0.0	2.1	0.0	0.9	0.0	0.8	0.0	0.9	0.0	2.2	0.0	1.0	0.0	1.0	0.0	1.0
	D_I10_Q5_C2	0.0	2.4	0.0	0.5	0.0	-0.4	0.0	-0.2	0.0	2.6	0.0	0.9	0.0	0.7	0.0	0.8
	D_I10_Q5_C3	0.0	1.2	0.0	0.3	0.0	-0.6	0.0	-1.0	0.0	1.9	0.0	0.8	0.0	0.9	0.0	0.7
	D_I10_Q5_C4	0.0	3.0	0.0	0.6	0.0	-0.3	0.0	-0.5	0.0	3.4	0.0	0.9	0.0	0.5	0.0	0.2
	D_I10_Q5_C5	0.0	2.6	0.0	0.7	0.0	0.3	0.0	0.7	0.0	2.9	0.0	0.9	0.0	0.7	0.0	0.6
	D_I10_Q10_C1	0.0	1.6	0.0	1.4	0.0	1.3	0.0	1.5	0.0	1.8	0.0	1.6	0.0	1.8	0.0	1.9
	D_I10_Q10_C2	0.0	2.9	0.0	3.0	0.0	0.9	0.0	0.6	0.0	2.6	0.0	3.3	0.0	2.2	0.0	2.3
	D_I10_Q10_C3	0.0	1.7	0.0	1.6	0.0	1.0	0.0	1.2	0.0	1.8	0.0	1.7	0.0	1.2	0.0	1.4
	D_I10_Q10_C4	0.0	3.3	0.0	3.4	0.0	11.3	0.0	2.0	0.0	3.5	0.0	3.6	0.0	12.2	0.0	2.2
	D_I10_Q10_C5	0.0	3.5	0.0	3.6	0.0	3.1	0.0	1.9	0.0	3.6	0.0	3.5	0.0	3.9	0.0	2.2
$I30$	D_I30_Q5_C1	0.0	-1535.8	0.0	-1644.5	0.0	-3587.9	0.0	-117.3	0.0	5.3	0.0	19.2	0.0	-358.3	0.0	30.2
	D_I30_Q5_C2	0.0	-3546.9	0.0	-3420.7	0.9	-3084.4	0.0	11.6	0.0	42.7	0.0	164.3	0.9	381.6	0.0	25.8
	D_I30_Q5_C3	0.0	-2359.7	0.0	-3560.8	2.8	-3115.6	0.0	-194.5	0.0	-0.0	0.0	14.9	2.8	472.9	0.0	25.2
	D_I30_Q5_C4	-3.2	-3578.6	0.0	-3564.7	4.3	-3540.3	0.0	-67.0	0.0	3.7	0.0	15.9	4.3	-15.8	0.0	26.4
	D_I30_Q5_C5	0.0	-1897.7	0.0	-3562.7	0.0	-3581.2	0.0	19.7	0.0	5.7	0.0	-6.2	0.0	-54.0	0.0	25.8
	D_I30_Q10_C1	0.0	12.2	0.0	9.0	3.6	-542.5	0.0	-1603.9	0.0	23.9	0.0	38.5	3.6	30.7	0.0	62.3
	D_I30_Q10_C2	0.0	-47.4	0.0	-224.1	3.0	-1536.3	0.0	-335.8	0.0	16.8	0.0	50.0	3.0	83.3	0.0	18.2
	D_I30_Q10_C3	0.0	-6.3	0.0	-49.5	0.0	-476.7	0.0	-3582.2	0.0	18.6	0.0	32.9	0.0	14.0	0.0	18.3
	D_I30_Q10_C4	0.0	-153.7	0.0	-699.2	0.0	-3552.1	0.0	-3579.6	0.0	18.8	0.0	34.6	0.0	42.5	0.0	16.0
	D_I30_Q10_C5	0.0	11.5	0.0	4.9	5.0	-343.0	0.0	-3579.7	0.0	18.0	0.0	36.8	5.0	26.7	0.0	18.3
	D_I30_Q30_C1	0.0	26.4	0.0	26.7	0.0	20.4	0.0	51.1	0.0	27.1	0.0	29.2	0.0	29.5	0.0	56.5
	D_I30_Q30_C2	0.0	18.2	0.0	28.0	0.0	31.0	0.0	17.9	0.0	18.9	0.0	29.4	0.0	40.9	0.0	21.4
	D_I30_Q30_C3	0.0	20.3	0.0	26.5	0.0	23.5	0.0	16.7	0.0	20.8	0.0	29.1	0.0	30.6	0.0	21.4
	D_I30_Q30_C4	0.0	25.3	0.0	28.2	0.0	36.0	0.0	18.0	0.0	26.0	0.0	29.4	0.0	43.3	0.0	21.0
	D_I30_Q30_C5	0.0	26.8	0.0	24.9	0.0	29.1	0.0	14.7	0.0	27.6	0.0	27.0	0.0	37.1	0.0	20.6
$I50$	D_I50_Q5_C1	-2.2	-3589.5	-2.2	-3553.2	0.6	-3560.8	0.0	-56.7	0.0	-1985.0	0.0	-1555.3	0.6	-2580.4	0.0	40.1
	D_I50_Q5_C2	0.0	-3588.5	0.0	-3585.8	-0.5	-3559.5	0.0	-602.6	0.0	-1212.5	0.0	-1594.3	0.0	-3558.6	0.0	49.6
	D_I50_Q5_C3	0.0	-3570.7	0.0	-3546.2	-0.5	-3533.1	0.0	-43.0	0.0	-149.0	0.0	-169.7	0.0	-1207.3	0.0	47.9
	D_I50_Q5_C4	0.0	-3543.7	0.0	-3507.9	0.0	-3526.2	0.0	-74.1	0.0	-141.5	0.0	-236.4	0.0	-1140.9	0.0	37.2
	D_I50_Q5_C5	0.0	-3575.0	0.0	-3556.1	0.0	-3539.7	0.0	-72.1	0.0	-40.0	0.0	-109.5	0.0	-149.3	0.0	43.7
	D_I50_Q10_C1	0.0	-3553.9	0.0	-3536.1	2.1	-3580.9	0.0	-14.0	0.0	30.9	0.0	42.6	2.1	-30.3	0.0	36.0
	D_I50_Q10_C2	0.0	-2416.9	0.0	-1838.7	0.0	-3534.9	0.0	-3.4	0.0	62.9	0.0	61.1	0.0	57.8	0.0	43.0
	D_I50_Q10_C3	0.0	-409.1	0.0	-3545.2	3.9	-3464.7	0.0	-158.2	0.0	52.5	0.0	40.8	3.9	127.9	0.0	61.9
	D_I50_Q10_C4	0.0	-3535.3	0.0	-3555.2	0.0	-3550.5	0.0	-224.8	0.0	54.8	0.0	28.4	0.0	36.6	0.0	43.4
	D_I50_Q10_C5	0.0	-2828.6	0.0	-3551.6	4.5	-3436.8	0.0	-98.8	0.0	43.2	0.0	43.6	4.5	147.8	0.0	33.4
	D_I50_Q50_C1	0.0	27.8	0.0	64.9	0.0	44.2	0.0	59.6	0.0	28.9	0.0	68.2	0.0	57.6	0.0	72.2
	D_I50_Q50_C2	0.0	19.4	0.0	80.4	0.0	64.8	0.0	61.4	0.0	20.6	0.0	82.7	0.0	81.0	0.0	70.0
	D_I50_Q50_C3	0.0	28.5	0.0	61.5	0.0	46.8	0.0	54.5	0.0	29.6	0.0	63.9	0.0	67.4	0.0	63.4
	D_I50_Q50_C4	0.0	17.9	0.0	65.6	0.0	61.2	0.0	43.5	0.0	19.6	0.0	68.2	0.0	75.4	0.0	52.5
	D_I50_Q50_C5	0.0	17.1	0.0	78.9	0.0	29.1	0.0	49.9	0.0	18.9	0.0	80.7	0.0	46.2	0.0	54.2

Table D.4.1: Summary heuristic comparison for the A-MSSP formulations (D-instances).

	Instance	AS								AS^c							
		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$		$\beta = 0.2$		$\beta = 0.5$		$\beta = 0.8$		$\beta = 1.0$	
		$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$	$HGap$	$HTime$
	$U\_I10\_Q5\_C1$	0.0	1.7	0.0	0.2	0.0	-0.8	0.0	0.8	0.0	2.1	0.0	0.8	0.0	0.1	0.0	0.4
	$U\_I10\_Q5\_C2$	0.0	2.0	0.0	-0.2	0.0	-0.8	0.0	-0.8	0.0	1.5	0.0	-0.0	0.0	-0.8	0.0	-0.8
	$U\_I10\_Q5\_C3$	0.0	1.9	0.0	0.5	0.0	-0.9	0.0	-0.7	0.0	2.2	0.0	0.6	0.0	-0.7	0.0	-0.8
	$U\_I10\_Q5\_C4$	0.0	2.2	0.0	-0.4	0.0	-0.9	0.0	-0.7	0.0	2.4	0.0	0.1	0.0	-0.8	0.0	-0.8
	$U\_I10\_Q5\_C5$	0.0	2.0	0.0	0.4	0.0	-0.6	0.0	-0.0	0.0	2.2	0.0	0.5	0.0	-0.0	0.0	0.8
$I10$																	
	$U\_I10\_Q10\_C1$	0.0	2.2	0.0	1.7	0.0	-0.1	0.0	1.5	0.0	2.4	0.0	2.1	0.0	1.9	0.0	1.4
	$U\_I10\_Q10\_C2$	0.0	3.5	0.0	1.6	0.0	-6.1	0.0	1.5	0.0	3.0	0.0	1.9	0.0	-1.6	0.0	2.5
	$U\_I10\_Q10\_C3$	0.0	3.3	0.0	2.9	0.0	-0.3	0.0	1.6	0.0	3.2	0.0	3.4	0.0	2.1	0.0	1.9
	$U\_I10\_Q10\_C4$	0.0	3.0	0.0	2.8	0.0	-4.7	0.0	1.8	0.0	0.8	0.0	3.0	0.0	-7.8	0.0	2.2
	$U\_I10\_Q10\_C5$	0.0	2.6	0.0	2.6	0.0	10.4	0.0	0.5	0.0	2.7	0.0	3.0	0.0	14.8	0.0	2.0
	$U\_I30\_Q5\_C1$	0.0	-209.1	0.0	-493.4	2.8	-3566.5	0.0	20.2	0.0	-3.1	0.0	17.6	2.8	-344.4	0.0	23.2
	$U\_I30\_Q5\_C2$	0.0	-1844.6	0.0	-1990.5	4.7	-3482.2	0.0	21.7	0.0	-132.9	0.0	-249.4	4.7	-592.0	0.0	25.1
	$U\_I30\_Q5\_C3$	0.0	-549.3	0.0	-889.2	2.3	-3549.2	0.0	31.8	0.0	-95.3	0.0	-41.5	2.3	-851.5	0.0	55.1
	$U\_I30\_Q5\_C4$	0.0	-385.6	0.0	-404.7	0.0	-3574.5	0.0	-79.9	0.0	2.0	0.0	37.2	0.0	-92.6	0.0	32.5
	$U\_I30\_Q5\_C5$	0.0	-167.1	0.0	-1476.5	5.6	-3554.1	0.0	25.3	0.0	7.4	0.0	-24.6	5.6	-350.1	0.0	27.7
	$U\_I30\_Q10\_C1$	0.0	-3.2	0.0	-22.6	0.0	-805.9	0.0	-69.1	0.0	19.1	0.0	48.9	0.0	31.1	0.0	47.8
	$U\_I30\_Q10\_C2$	0.0	-61.0	0.0	-264.5	0.0	-1250.5	0.0	-199.3	0.0	4.3	0.0	43.3	0.0	-5.1	0.0	12.1
$I30$	$U\_I30\_Q10\_C3$	0.0	-16.1	0.0	-247.0	1.9	-1157.2	0.0	-501.5	0.0	13.7	0.0	31.1	1.9	3.4	0.0	18.4
	$U\_I30\_Q10\_C4$	0.0	-21.7	0.0	-274.2	5.4	-734.5	0.0	-230.1	0.0	16.8	0.0	122.4	5.4	14.6	0.0	7.2
	$U\_I30\_Q10\_C5$	0.0	-47.8	0.0	-162.9	5.3	-1234.3	0.0	-1673.9	0.0	15.1	0.0	43.3	5.3	-1.6	0.0	-15.4
	$U\_I30\_Q30\_C1$	0.0	32.1	0.0	42.1	0.0	-177.9	0.0	22.6	0.0	34.7	0.0	95.1	0.0	62.5	0.0	26.8
	$U\_I30\_Q30\_C2$	0.0	12.1	0.0	-59.0	0.0	-540.0	0.0	26.6	0.0	15.7	0.0	42.1	0.0	9.0	0.0	35.3
	$U\_I30\_Q30\_C3$	0.0	27.4	0.0	45.1	0.0	-504.2	0.0	60.2	0.0	30.4	0.0	53.2	0.0	28.9	0.0	65.1
	$U\_I30\_Q30\_C4$	0.0	13.5	0.0	36.8	5.4	-251.1	0.0	47.1	0.0	16.5	0.0	47.7	5.4	21.5	0.0	54.3
	$U\_I30\_Q30\_C5$	0.0	12.1	0.0	-35.2	5.4	-363.2	0.0	48.1	0.0	15.5	0.0	38.4	5.4	14.0	0.0	56.6
	$U\_I50\_Q5\_C1$	0.0	-461.2	0.0	-741.7	0.0	-3590.5	0.0	-95.7	0.0	187.3	0.0	28.1	0.0	-3590.3	0.0	68.1
	$U\_I50\_Q5\_C2$	0.0	-1602.0	0.0	-1492.8	3.3	-3591.8	0.0	-13.8	0.0	-15.7	0.0	-65.5	3.3	-3592.2	0.0	40.5
	$U\_I50\_Q5\_C3$	0.0	-3515.9	0.0	-3479.1	-2.6	-3598.6	0.0	-205.8	0.0	-3515.9	0.0	-3479.1	0.0	-622.5	0.0	40.0
	$U\_I50\_Q5\_C4$	0.0	-3496.0	0.0	-3511.4	-2.7	-3593.3	0.0	-62.5	0.0	-3496.3	0.0	-3510.9	0.0	-466.2	0.0	24.8
	$U\_I50\_Q5\_C5$	-2.6	-3520.6	0.0	-2463.8	-0.7	-3587.5	0.0	-79.0	0.0	-914.7	0.0	-353.2	0.0	-614.0	0.0	32.8
	$U\_I50\_Q10\_C1$	0.0	-562.8	0.0	-924.4	0.0	-2484.4	0.0	-927.7	0.0	27.1	0.0	5.9	0.0	117.0	0.0	43.5
	$U\_I50\_Q10\_C2$	0.0	-1362.1	0.0	-3502.9	5.0	-3416.1	0.0	-833.5	0.0	17.1	0.0	-57.1	5.0	-2.2	0.0	43.8
$I50$	$U\_I50\_Q10\_C3$	0.0	-554.8	0.0	-1413.3	3.8	-3465.4	0.0	-204.0	0.0	50.6	0.0	63.1	3.8	-125.7	0.0	63.4
	$U\_I50\_Q10\_C4$	0.0	-1346.8	0.0	-2290.2	2.6	-3446.0	0.0	-694.3	0.0	25.4	0.0	28.3	2.6	17.7	0.0	24.5
	$U\_I50\_Q10\_C5$	0.0	-1251.7	0.0	-2559.1	-2.4	-3482.3	0.0	-327.6	0.0	-15.3	0.0	-44.2	0.0	-251.0	0.0	53.8
	$U\_I50\_Q50\_C1$	0.0	29.3	0.0	124.0	1.8	-913.0	0.0	81.1	0.0	38.4	0.0	159.5	1.8	124.4	0.0	99.2
	$U\_I50\_Q50\_C2$	0.0	31.2	0.0	45.3	5.0	-2312.0	0.0	47.4	0.0	39.9	0.0	96.9	5.0	143.5	0.0	69.2
	$U\_I50\_Q50\_C3$	0.0	33.4	0.0	63.1	0.0	-528.1	0.0	121.4	0.0	40.8	0.0	226.9	0.0	284.1	0.0	135.4
	$U\_I50\_Q50\_C4$	0.0	31.0	0.0	280.2	0.0	-1631.6	0.0	181.3	0.0	40.2	0.0	345.9	0.0	170.4	0.0	194.6
	$U\_I50\_Q50\_C5$	0.0	31.7	0.0	220.4	5.1	-862.7	0.0	122.1	0.0	40.1	0.0	249.5	5.1	309.6	0.0	145.6

Table D.4.2: Summary heuristic comparison for the A-MSSP formulations (U-instances).