



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## Scheduling Policies for Multi-Period Services

**Citation for published version:**

Nunez-del-Toro, C, Fernandez, E, Kalcsics, J & Nickel, S 2016, 'Scheduling Policies for Multi-Period Services', *European Journal of Operational Research*, vol. 251, no. 3, pp. 751-770.  
<https://doi.org/10.1016/j.ejor.2015.12.002>

**Digital Object Identifier (DOI):**

[10.1016/j.ejor.2015.12.002](https://doi.org/10.1016/j.ejor.2015.12.002)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Peer reviewed version

**Published In:**

European Journal of Operational Research

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Scheduling Policies for Multi-Period Services

Cristina Núñez-del-Toro<sup>a,\*</sup>, Elena Fernández<sup>a</sup>, Jörg Kalcsics<sup>b</sup>, Stefan Nickel<sup>c</sup>

<sup>a</sup>*Statistics and Operations Research Department, Universitat Politècnica de Catalunya, Barcelona, Spain*

<sup>b</sup>*School of Mathematics, University of Edinburgh, Edinburgh, Scotland*

<sup>c</sup>*Institute of Operations Research, Karlsruhe Institute of Technology, Karlsruhe, Germany*

---

## Abstract

This paper discusses a multi-period service scheduling problem. In this problem, a set of customers is given who periodically require service over a finite time horizon. To satisfy the service demands, a set of operators is given, each with a fixed capacity in terms of the number of customers an operator can serve per period. The task is to determine for each customer the periods in which he will be visited by an operator such that the periodic service requests of the customers are adhered to and the total number of operators used over the time horizon is minimal. Two alternative policies for scheduling customer visits are considered. In the first one, a customer is visited just on time, i.e., in the period where he or she has a demand for service. The second policy allows service visits ahead of time. The rationale behind this policy is that allowing irregular visits may reduce the overall number of operators needed throughout the time horizon. To solve the problem, integer linear programming formulations are proposed for both policies and numerical experiments are presented that show the reduction in the number of operators used when visits ahead of time are allowed. As only small instances can be solved optimally, a heuristic algorithm is introduced in order to obtain good quality solutions and shorter computing times.

*Keywords:* combinatorial optimization, multi-period problems, service scheduling, heuristics

---

## 1. Introduction

In this paper we discuss a problem that is derived from an application arising in the context of collection and recycling of waste electrical and electronic equipment (WEEE). According to EU regulations, inhabitants can return their WEEE free of charge at collection stations which are run

---

\*Corresponding author

*Email address:* cristina.nunez@upc.edu (Cristina Núñez-del-Toro)

by the local municipalities. Once a storage container at a station is full, one of the companies selling electrical or electronic products is responsible for picking up the container and recycling its contents, see for example Queiruga et al. [21], Fernández et al. [12]. As most companies do not want or do not have the capabilities to do this themselves, they commission logistics providers to carry out these tasks. At the stations, different product categories are collected in different types of containers, e.g., household appliances are collected in iron-barred boxes. As 5-10 boxes fit on a truck, the logistics provider can schedule a truck to visit several collection stations on a day. As the filling rate of these boxes differs between stations, one should organize individual schedules that allow for irregular visits to stations and aim at maximizing vehicle utilizations. Thus, the task is to decide for each collection station in which periods a box should be picked such that no box overflows and as few trucks as possible have to be deployed. Focussing on the tactical modeling aspects of reducing the number of used vehicles 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 [22].

The rest of the paper is organized as follows. In the next section we formally describe our problem and its modelling assumptions. In Section 3, we review the relevant literature. In Section 4 we propose mathematical formulations for two different collection policies. The effect of each policy is analyzed in Section 5 where we derive some insights into the modeling aspects of the problem. As we can only solve small instances optimally within a reasonable amount of time, we introduce in Section 6 a heuristic for the problem. In Section 7 we present computational results to underline the efficiency of the heuristic. The paper ends with some conclusions.

## 2. Problem Description

The collection problem described above falls within a more general class of multi-period service scheduling problems (MSSP). In the following, we briefly outline the main components of the latter to obtain a generic, application independent description of our problem. We are given a set of *customers* (e.g., collection stations) who have periodically recurring demand for some type of service (e.g., emptying boxes), a set of *operators* (e.g., trucks) who can deliver this service, and a planning horizon that is partitioned into a set of time periods of equal length (e.g., days). The index set of customers, operators, and time periods is denoted by  $I$ ,  $K$ , and  $T$ , respectively. Each

time a customer has demand for service, we call this a *service request*. Typically, no request spans more than one period and no customer has more than one request per period. The periods in which customers have service requests can either be given a priori or they can be generated dynamically based on satisfied requests, that is, each time a customer receives service in a period this triggers a new request. Such a dynamically generated request can either be open-ended or have a *due date* until which it has to be satisfied. In the latter case, the request can be satisfied exactly in the period where it is due, but may also be serviced in a period prior to the due date. The operators or servers can provide the service at the customer or the operator location, or remotely. 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 carries out a *service visit*. Every operator has a fixed capacity per time period and each service visit consumes a certain amount of the capacity. Moreover, a fixed cost may be charged whenever an operator attends to one or more customers in a period. Additionally, the satisfaction of a request may incur a fixed and/or variable cost, for example depending on the time elapsed since the last service period. The task is then to decide for each customer in which periods to schedule a service visit such that all customers receive their desired service, possibly meeting the due dates, and the operator capacities are adhered to. Concerning the optimization goals, typically a schedule is sought that minimizes the overall costs or the maximal number of operators required in a period or maximizes the regularity of the service schedules.

In our problem, service requests are generated dynamically with due dates. The due date for the next request is hereby derived from the current service period and the expected number of periods  $s_i \in \mathbb{N}$  it takes for a box to be filled (which is known for each customer and independent from time).  $s_i$  is also called the *service interval* of  $i$ . For each customer, the first service request is generated immediately before the start of the planning horizon, i.e., in time period 0. A *service cluster* consists of a set of customers visited by the same operator in the same period. Because different customers may have different service intervals and we do not assume that regular schedules should apply, the number of operators to be used along the time horizon is established on a per period basis. Hence, 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.

The maximal number of customers  $Q$  an operator can serve per period is fixed and identical for each operator, i.e., operators have capacity  $Q$  and each service visit consumes one unit of capacity.

Each service request can be satisfied by any operator and a customer can be visited by a different operator each time. Whenever an operator is used in a period, a fixed cost is incurred which is identical for each operator and period. Moreover, no fixed or variable costs are charged for service visits.

We consider two different scheduling policies. In the first one, we schedule the next service visit as late as possible, i.e., in the period where the current request is due. We call this an *on time* visit and the corresponding strategy a *Periodic Service policy* (PS). 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). The goal is then to minimize the total number of operators used over all periods of the time horizon. We call this the *Periodic Multi-period Service Scheduling Problem*, for short Periodic MSSP or P-MSSP. An ordered set of service periods  $C_i = \{t_i^1, \dots, t_i^{|C_i|}\} \subseteq T$  is called a *periodic calendar* for customer  $i$  if  $t_i^1 \leq s_i$  and the number of periods between any two consecutive elements in  $C_i$  equals the service interval  $s_i$ . Even if the first service period of a customer uniquely determines all subsequent service periods, this policy still does not necessarily produce identical service clusters because customers may have differing service intervals. In the second policy, we assume that a customer may be visited *ahead of time*, i.e., before the due date of the service request. We call this an *Aperiodic Service policy* (AS). In this policy, the number of periods between consecutive service periods  $t_i^k$  and  $t_i^{k+1}$  for customer  $i$  may vary from time to time. 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. An ordered set of service periods  $C_i = \{t_i^1, \dots, t_i^{|C_i|}\} \subseteq T$  is called an *aperiodic calendar* for customer  $i$  if  $t_i^1 \leq s_i$  and the number of periods between any two consecutive elements in  $C_i$  does not exceed the service interval  $s_i$ . If  $t_i^{k+1} - t_i^k$  is smaller than  $s_i$  we call  $s_i - (t_i^{k+1} - t_i^k)$  the *earliness* of the visit. The earliness of an aperiodic calendar  $C_i$  is the total earliness of all visits scheduled in the calendar, i.e.,  $\sum_{k=1}^{|C_i|-1} (s_i - t_i^{k+1} + t_i^k) = (|C_i| - 1) s_i - t_i^{|C_i|} + t_i^1$ , and the earliness of a customer is the earliness of his calendar. The goal is then to minimize a weighted sum of the total number of periods in which each operator is used and the total earliness of all customers. We call this the *Aperiodic Multi-period Service Scheduling Problem*, for short Aperiodic MSSP or A-MSSP. Note that the earliness of a service visit can be interpreted as a variable cost for satisfying the service request.

In the remainder, we assume  $T$  to be finite. Moreover, we assume that all customers have been serviced just before the start of the planning horizon and that these service visits do not influence the periodic calendars. Next, we illustrate the differences between the policies and highlight the utility of the AS policy.

**Example 1.** 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  $s_1 = 2$ ,  $s_2 = 3$ ,  $s_3 = 4$  and  $s_4 = 3$  periods, respectively. An operator can serve  $Q = 3$  customers per period. Figure 1 displays the schedule of an optimal solution if the PS policy is applied. For example, we have  $C_4 = \{3, 6, 9, 12\}$ . 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 nine operators over the twelve weeks.

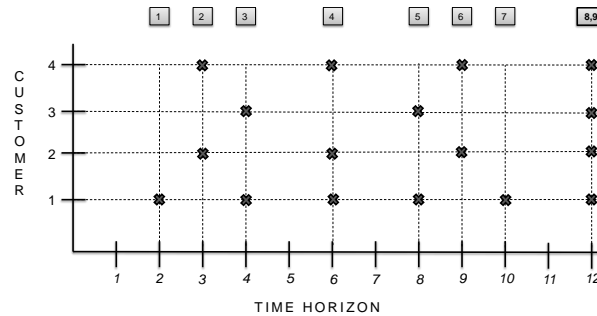


Figure 1: Solution of the example for the periodic service policy

However, under the AS policy we just need eight operators instead of nine, see Figure 2. We can 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.

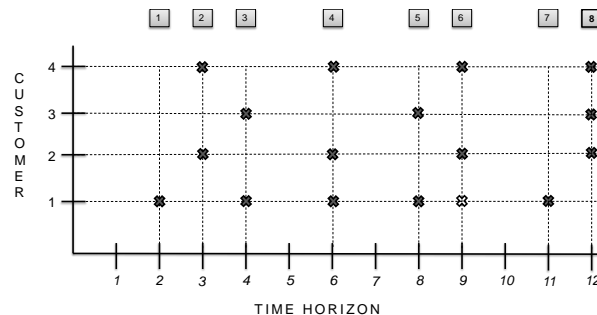


Figure 2: Solution of the example for aperiodic service policy

A main aim of this paper is to compare the two different service policies. We are especially interested 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 that visits are scheduled ahead of time. To that end we propose mathematical programming formulations to solve the problems optimally and we develop a heuristic solution approach.

### 3. Literature Review

In the following we only review literature on problems where scheduling services is the one and only decision. There are many other applications from very diverse fields where service scheduling appears as a component, e.g., in periodic vehicle routing problems (Campbell and Wilson [8]) or inventory routing problems (Coelho et al. [9]) when flexible visit frequencies are allowed. However, the additional decisions to be made in these applications typically result in considerably different problems, requiring different modelling approaches, and result in different solutions. For example minimizing the length of vehicle tours does not necessarily minimize the total number of tours, i.e., the number of operators, as can be seen with small examples.

A closely related application deals with inspections of technical equipment, like machines or airplanes, where technicians regularly carry out preventive service and maintenance. There is a fixed cost for carrying out an inspection and a variable operating cost per period that is proportional to the number of periods since the last inspection. The goal is to determine an inspection schedule minimizing the total average costs over an infinite planning horizon (Anily et al. [1], Bar-Noy et al. [2], Grigoriev et al. [15]). In our terminology, customers coincide with machines and operators with technicians where each technician can maintain only one machine per period. Service requests are generated dynamically but usually without due dates. The operating costs per period can be equivalently modelled by means of the variable costs for service requests. [2] show that the problem is NP-hard and present several approximation algorithms for the case where a fixed number of machines can be inspected in each period. [1] and [15] consider instead the case of just a single operator. The former derive conditions for the existence of optimal solutions and propose an exact, but non-polynomial solution method. They also derive lower bounds and a bounded error heuristic. The latter present different linear and non-linear programming formulations for the problem as well as an exact solution approach using column generation. Similar problems occur in machine replacement (McClurg and Chand [20]) and sensor scheduling (Yavuz and Jeffcoat [24]). Instead of minimizing costs, some authors try to derive regular schedules by minimizing

the variability between consecutive service periods for customers. Corominas et al. [10] present a linear programming formulation and introduce several improvements for it. In subsequent papers, García-Villoria et al. [14] and Corominas et al. [11] present a specially tailored branch-and-bound algorithm and various heuristics, respectively. The same objective is employed in Herrmann [17] in the context of periodically collecting waste from rooms in a health care facility. Apart from these “physical” applications, there are also scheduling problems referring to maintenance and backups of computer systems.

Another related problem 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 (Bar-Noy and Ladner [3], Bar-Noy et al. [4, 5]). This problem is called windows scheduling. Operators coincide with channels, pages with customers, and page transmissions with service requests. The latter are generated dynamically with due dates. Capacities and consumption rates are identical to one. There are no costs involved and the objective is to minimize the maximum number of operators used in a period. [3] present an algorithm to construct asymptotically close to optimal schedules where each page has a unit transmission time. [5] 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. Finally, [4] establishes a relation between window scheduling and bin packing. Similar problems occur in media-on-demand systems.

Also related are replenishment problems in vendor managed inventory systems under the assumption of direct deliveries. Each retailer faces a constant demand and no shortages are allowed. Inventory is replenished from a central distribution center and a vehicle can just visit one retailer per period. In our context, customers correspond to retailers, operators to vehicles, and service requests to stock replenishment. The latter are generated dynamically with due dates. Capacities and consumption rates are identical to one. Campbell and Hardin [7] assume that all visits have to be on time and try to minimize the maximal number of vehicles needed over all periods. They derive several complexity results and propose a greedy heuristic. They also look at the cases where just one vehicle is available and where the capacity consumption is strictly smaller than one, i.e., more than one retailer can be replenished by a vehicle per period. The same restrictions and goals



are used in Xu et al. [23], except that service visits can also be ahead of time. They derive an approximation algorithm to solve the problem. Instead of minimizing the fleet size, Gallego and Simchi-Levi [13], Bertazzi [6] consider the total inventory holding, transportation, and fixed ordering costs. They compare delivery tours versus direct deliveries and the effect of different replenishment policies under direct deliveries, respectively.

Other related problems occur in task scheduling and periodic assignment problems. Han et al. [16] consider the problem of scheduling tasks on a single machine or processor where distance constraints specify the maximal time between to consecutive executions of a task, i.e., service request are generated dynamically with a maximum duration. They derive feasibility aspects and schedulability conditions. Korst et al. [19] discuss the problem of assigning periodic operations to processors. A processor can just work on one job per period. Service requests are again generated dynamically with a maximum duration, but no ahead of time processing is allowed. The goal is to minimize the maximum number of processors needed. 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 mounted 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 and a reasonable goal is to minimize the total number of in the field periods of trucks or teams (cf. Jeffries and O’Hanley [18]). In its most general form, this problem can also be viewed as a multi-period bin packing problem in which items (customers) must be assigned to bins (operators) where each bin refers to a certain time period. The goal is then to minimize the number of bins.

While all of the above mentioned problems share common components with our multi-period service scheduling problem, neither of them matches exactly. They either differ in the objective function, in the way service requests have to be satisfied, or in the definition of operator capacities/consumption rates.

#### 4. Mathematical Formulations

In this section we present mixed integer linear programming formulations for the two service policies. In the MSSP the following decisions must be made:

- Determine a (periodic/apperiodic) calendar for each customer, i.e., the set of periods when he

will be served.

- Assign each service period of a customer to an operator taking into account the operator capacities.

To that end, we define the following sets of decision variables for  $i \in I$ ,  $k \in K$ , and  $t \in T$ :

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

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

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

Before we present the formulations for the periodic and aperiodic MSSP, recall our assumption that all customers have been serviced just before the start of the planning horizon and that these service visits do not influence the periodic calendars.

#### 4.1. Periodic Service Policy for the Multi-Period Service Scheduling Problem

The P-MSSP can be formulated as:

$$(PS) \quad \min \sum_{t \in T} \sum_{k \in K} z_k^t \tag{1}$$

$$s.t. \quad \sum_{t=1}^{s_i} x_i^t = 1 \quad i \in I \tag{2}$$

$$x_i^t = x_i^{t+s_i} \quad i \in I, t \in \{1, \dots, |T| - s_i\} \tag{3}$$

$$x_i^t = \sum_{k \in K} y_{ik}^t \quad i \in I, t \in T \tag{4}$$

$$\sum_{i \in I} y_{ik}^t \leq Qz_k^t \quad k \in K, t \in T \tag{5}$$

$$Qz_k^t \leq \sum_{i \in I} y_{i,k-1}^t \quad k \in K \setminus \{1\}, t \in T \tag{6}$$

$$x_i^t, z_k^t, y_{ik}^t \in \{0, 1\} \quad i \in I, k \in K, t \in T \tag{7}$$

Objective (1) minimizes the total number of operators used in each period of the time horizon. The first service period for each customer is established by Constraints (2), whereas consecutive service periods throughout the time horizon are determined by Constraints (3). Constraints (4) guarantee that if a customer is served in period  $t$ , then he is assigned to some operator in that period. Constraints (5) are capacity constraints that ensure that the number of customers assigned to each operator must not exceed her capacity. Constraints (6) are symmetry breaking constraints imposing that in each period operator  $k$  is not used unless operators  $1, \dots, k-1$  are full, i.e., each of them has  $Q$  assigned customers. Note that although these constraints restrict the feasible domain, they have no effect on the optimal solution because we minimize the total number of operators. Finally, Constraints (7) enforce variables to be binary.

#### 4.2. Aperiodic Service Policy for the Multi-Period Service Scheduling Problem

In the AS policy, the first service period of a customer is not enough to determine his subsequent service periods over the time horizon. In particular, if customer  $i$  is visited in period  $t \leq |T| - s_i$ , we know that he must be visited again no more than  $s_i$  periods later, but we do not know the specific period when he will be serviced again within the time interval  $[t+1, t+s_i]$ . We introduce variables that identify the number of periods between two consecutive service visits for a given customer. In particular, for  $i \in I$ ,  $t \in \{0, \dots, |T| - s_i\}$ ,  $h \in \{1, \dots, s_i\}$  we define

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

Due to the fixed time horizon, we assume that service visits scheduled after period  $|T| - s_i$  will not generate new requests.

To reduce the negative effect that ahead of time visits may cause, we minimize not only the total number of operators used but also the total earliness of all customers using a convex combination

with parameter  $\beta \in [0, 1]$ . A first formulation that we propose for the A-MSSP is the following:

$$(AS_0) \quad \min \quad \beta \sum_{t \in T} \sum_{k \in K} z_k^t + (1 - \beta) \sum_{i \in I} \sum_{t=1}^{|T|-s_i} \sum_{h=1}^{s_i} (s_i - h) f_i^{th} \quad (8)$$

$$s.t. \quad \sum_{h=1}^{s_i} f_i^{0h} \geq 1 \quad i \in I \quad (9)$$

$$x_i^t \leq \sum_{h=1}^{s_i} x_i^{t+h} \quad i \in I, t \in \{0, \dots, |T| - s_i\} \quad (10)$$

$$x_i^t = \sum_{h=1}^{s_i} f_i^{th} \quad i \in I, t \in \{0, \dots, |T| - s_i\} \quad (11)$$

$$x_i^t + x_i^{t+h} \leq \sum_{s=1}^{h-1} x_i^{t+s} + f_i^{th} + 1 \quad i \in I, t \in \{0, \dots, |T| - s_i\}, h \in \{1, \dots, s_i\} \quad (12)$$

$$z_k^t \leq \sum_{i \in I} y_{ik}^t \quad k \in K, t \in T \quad (13)$$

$$x_i^t = \sum_{k \in K} y_{ik}^t \quad i \in I, t \in T \quad (14)$$

$$\sum_{i \in I} y_{ik}^t \leq Qz_k^t \quad k \in K, t \in T \quad (15)$$

$$Qz_k^t \leq \sum_{i \in I} y_{i,k-1}^t \quad k \in K \setminus \{1\}, t \in T \quad (16)$$

$$z_k^t, y_{ik}^t \in \{0, 1\} \quad i \in I, k \in K, t \in T \quad (17)$$

$$x_i^t \in \{0, 1\} \quad i \in I, t \in \{0, \dots, |T|\} \quad (18)$$

$$f_i^{th} \in \{0, 1\} \quad i \in I, t \in \{1, \dots, |T| - s_i\}, h \in \{1, \dots, s_i\} \quad (19)$$

Objective (8) minimizes a weighted sum of the total number of operators used over 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 (9) guarantee that the first service period for each customer occurs within his service interval. Constraints (10) are scheduling constraints, which ensure that the number of periods between two consecutive service periods for the same customer never exceeds his service interval. Constraints (11) and (12) relate the  $x$  and  $f$  variables, where the latter force the variable  $f_i^{th}$  to one, if customer  $i$  has been visited at periods  $t$  and  $t + h$  and there

is no other service period for  $i$  between  $t$  and  $t + h$ . Because we have a minimization problem, the variable  $f_i^{th}$  will be zero whenever possible. Hence, we do not have to enforce this. Note that by combining (9) and (11) we always have  $x_i^0 = 1$ . In contrast to the P-MSSP, we have to explicitly model the initial service request generated in period 0. Constraints (13) relate the  $z$  and  $y$  variables, ensuring that customers will only be assigned to active operators. Constraints (14)-(16) have the same meaning as Constraints (4)-(6) in formulation *PS*. Finally, Constraints (17)-(19) enforce variables to be binary.

#### 4.3. Improvements on the Formulation $AS_0$

In this section we formalize several statements which are quite intuitive but not always straightforward to prove. In particular, we observe that the following constraints are also valid for  $AS_0$ :

$$f_i^{t-h,h} \leq x_i^t \quad i \in I, t \in T, h = \max\{1, t - (|T| - s_i)\}, \dots, \min\{s_i, t\}.$$

That is, if we schedule the next service period after period  $t - h$  to be in period  $t - h + h = t$ , then the corresponding  $x$ -variable has to be one. The domain of  $h$  ensures that  $0 \leq t - h \leq |T| - s_i$  and thus that the corresponding  $f$ -variable is defined. A tighter form of these constraints is

$$\sum_{h=\max\{1, t-(|T|-s_i)\}}^{\min\{s_i, t\}} f_i^{t-h,h} = x_i^t \quad i \in I, t \in T. \quad (20)$$

We denote by  $\Omega_0$  the domain defined by constraints (9)-(19) and by  $\Omega_1$  the domain defined by (9), (11)-(20). Let also  $\bar{\Omega}_0$  and  $\bar{\Omega}_1$  denote the respective domains when the binary conditions on the variables are replaced by non-negativity constraints. That is:

$$\begin{aligned} \Omega_0 &= \{(x, y, z, f) : \text{satisfying (9) - (19)}\}, \\ \Omega_1 &= \{(x, y, z, f) : \text{satisfying (9), (11) - (20)}\}, \\ \bar{\Omega}_0 &= \{\mathbf{0} \leq (x, y, z, f) \leq \mathbf{1} : \text{satisfying (9) - (16)}\}, \\ \bar{\Omega}_1 &= \{\mathbf{0} \leq (x, y, z, f) \leq \mathbf{1} : \text{satisfying (9), (11) - (16), (20)}\}. \end{aligned}$$

where  $\mathbf{0}$  and  $\mathbf{1}$  are vectors of appropriate dimensions with 0's and 1's, respectively. Next we see that both  $\Omega_0$  and  $\Omega_1$  give equivalent formulations for the A-MSSP. In particular, the following

statements hold (the proofs can be found in Appendix A):

**Proposition 1.**  $\Omega_0 = \Omega_1$ .

**Proposition 2.**  $\bar{\Omega}_1 \subseteq \bar{\Omega}_0$ .

**Remark 1.** *The reverse of Proposition 2 is not true. Table 1 gives an example to illustrate that  $\bar{\Omega}_0 \not\subseteq \bar{\Omega}_1$ . In particular, Table 1 displays the values of the  $x$  and  $f$  variables for the solution of the LP-relaxation of  $AS_0$  for the instance with  $|I| = 1$ ,  $|T| = 13$ ,  $|K| = 1$ ,  $|Q| = 3$ ,  $s_1 = 3$  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, (20) is not satisfied. This implies that a solution in  $\bar{\Omega}_0$  is not necessarily contained in  $\bar{\Omega}_1$ .*

$i$	$t$	$h$	$t+h$	$x_i^t$	$f_i^{th}$	$x_i^{t+h}$
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 1:  $x$  and  $f$  values for the LP-relaxation solution of  $AS_0$ .

From now on, we will use the following formulation for the A-MSSP:

$$\begin{aligned}
 (AS) \quad & \min \beta \sum_{t \in T} \sum_{k \in K} z_k^t + (1 - \beta) \sum_{i \in I} \sum_{t=1}^{|T|-s_i} \sum_{h=1}^{s_i} (s_i - h) f_i^{th} \\
 & s.t. \quad (x, y, z, f) \in \Omega_1
 \end{aligned}$$

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

**Corollary 1.** *The LP-relaxation of  $AS$  is tighter than the LP-relaxation of  $AS_0$ .*

Computational tests show that the LP-relaxation of  $AS$  provides lower bounds that are on average 72% tighter than those of formulation  $AS_0$ . Similarly, the CPU times using  $AS$  are 94% shorter than those for  $AS_0$ .

## 5. Empirical Comparison of Policies

In this section we focus on the effect of the PS and AS policies and on the performance of the formulations presented in Section 4. All experiments have been run on an Intel Core i7-2600 3.4 GHz with 8 GB Ram and operating system Windows 7, 64 bit. Both formulations were coded in C++ and solved with IBM ILOG CPLEX 12.5 with a CPU time limit of one hour.

### 5.1. Data Generation

Since we are not aware of any benchmark instances for the MSSP, we randomly generated 90 problem instances with the following characteristics. For the number of customers we chose  $|I| \in \{10, 30, 50\}$ . The respective instances are denoted 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  $s_i$  of the customers, we considered two different settings. The first one relates the service intervals to the number of visits per month:  $s_i \in \{4, 7, 15\}$  (eight times, four times, and twice per month, respectively). The second one considers the intervals  $s_i \in \{4, 5, \dots, 15\}$ . We abbreviate the two settings by  $D$  and  $U$ , respectively. In view of the “physical” services mentioned in the beginning, the capacity  $Q$  of the operators is 5 or 10. Moreover, to reflect the “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.

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 aperiodic policy, we consider four different values for  $\beta \in \{0.2, 0.5, 0.8, 1.0\}$ . The extreme value of  $\beta = 1.0$  was considered to analyze the effect of minimizing the total number of operators

used over all periods without penalizing earliness. This yields a total of 450 instances, 90 for the PS policy and 360 for the AS policy.

### 5.2. Comparison of the Formulations

For each  $|I|$  and  $Q$ , Table 2 shows the aggregated results obtained by CPLEX for the formulations *PS* and *AS* for the D- and U-instances. For *AS*, the results are presented for every value of  $\beta$ . The detailed results can be found in Tables B.1 and B.2 in Appendix B. Each table displays the total number of operators used and the total earliness over the five instances of the best solutions found ( $z$  and  $e$ , respectively). Columns labeled *Gap* show the average relative deviations in percent of the best-known solutions with respect to lower bounds at termination. The average CPU times in seconds and the number of optimally solved instances are given in columns *Time* and *Opt*, respectively. CPLEX was able to find at least one feasible solution for all instances within the time limit of one hour.

		<i>PS</i>																				<i>AS</i>																			
		$\beta = 0.2$					$\beta = 0.5$					$\beta = 0.8$					$\beta = 1.0$																								
		<i>z</i>	<i>Gap</i>	<i>Time</i>	<i>Opt</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>Opt</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>Opt</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>Opt</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>Opt</i>																
D-instances																																									
<i>I10</i>	<i>Q5</i>	51	0.0	0.8	5	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																
	<i>Q10</i>	40	0.0	0.1	5	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																
<i>I30</i>	<i>Q5</i>	140	0.0	1567.6	5	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																
	<i>Q10</i>	79	0.0	19.2	5	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																
	<i>Q30</i>	50	0.0	0.1	5	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																
<i>I50</i>	<i>Q5</i>	243	4.4	3601.9	0	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																
	<i>Q10</i>	132	2.0	1383.9	4	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																
	<i>Q50</i>	50	0.0	0.1	5	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																
U-instances																																									
<i>I10</i>	<i>Q5</i>	55	0.0	0.5	5	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																
	<i>Q10</i>	55	0.0	0.2	5	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																
<i>I30</i>	<i>Q5</i>	109	0.0	324.4	5	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																
	<i>Q10</i>	71	0.0	11.3	5	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																
	<i>Q30</i>	70	0.0	0.2	5	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																
<i>I50</i>	<i>Q5</i>	186	1.7	2355.2	2	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																
	<i>Q10</i>	99	0.0	362.8	5	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																
	<i>Q50</i>	75	0.0	0.4	5	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																

Table 2: Summary CPLEX results for the D- and U-instances with the MSSP formulations *PS* and *AS*.

For formulation *PS*, optimal solutions were obtained for 85% and 93% of the D-instances and U-instances, respectively. For formulation *AS*, the number of optimal solutions found depends on the selection of the parameter  $\beta$ . For  $\beta = 0.2$ , optimality of the best solution found could be proven for 78% and 93% of the instances. As the value of  $\beta$  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. Observe, however, that 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  $\beta$  and even higher than for formulation *PS*.

The computing times reinforce this situation. For formulation *PS*, the average values for the CPU time are 821.7 and 381.9 seconds for the D- and U-instances, respectively. For formulation *AS*, this average value also depends on the selection of the parameter  $\beta$ . For  $\beta = 0.2$ , the average values for the computing time are 1110.1 and 557.1 seconds. As the value of  $\beta$  increases, the higher the CPU time. The highest average values of CPU time correspond to  $\beta = 0.8$  with 1521.5 and 1699.0 seconds. Observe, however, that the average computing time for  $\beta = 1.0$  is substantially lower 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 CPU time might be due to the fact that we do not penalize earliness in the objective function for  $\beta = 1.0$  in contrast to  $\beta = 0.8$ .

As the earliness has no impact on the objective function for  $\beta = 1.0$ , the total earliness of the solutions is uncharacteristically high. To obtain a better picture on the amount of earliness and to make the numbers comparable to the cases  $\beta < 1.0$ , we have run a second round of experiments, called phase II, with the following characteristics. We solved each instance again but set  $\beta = 0$  and added the constraint  $\sum_{t \in T} \sum_{k \in K} z_k^t = z_{\beta=1}$  which fixes the total number of operators to be used to the number of operators,  $z_{\beta=1}$ , previously obtained for  $\beta = 1.0$ . The results are shown in Table 3. Although the amount of earliness increases very much from  $\beta = 0.8$  to  $\beta = 1.0$ , the total number of required operators often decreases considerably. Concerning the CPU times for phase II, we observe a similar pattern as for  $\beta < 1.0$ .

		D-instances				U-instances			
		<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>
<i>I10</i>	<i>Q5</i>	43	22	0.0	7.3	39	72	0.0	26.0
	<i>Q10</i>	29	150	0.0	3.0	30	142	0.0	9.1
<i>I30</i>	<i>Q5</i>	130	82	100.0	3602.2	106	8	60.0	2931.1
	<i>Q10</i>	70	77	15.0	2050.5	55	136	0.0	1779.1
	<i>Q30</i>	35	729	0.0	76.5	34	582	0.0	470.7
<i>I50</i>	<i>Q5</i>	234	47	100.0	3602.1	183	24	60.0	3060.7
	<i>Q10</i>	119	103	100.0	3602.0	92	138	100.0	3600.2
	<i>Q50</i>	35	1014	0.0	244.5	35	887	0.0	1982.6

Table 3: Summary of the CPLEX phase II results for formulation *AS*.

The optimality gap also underlines the difficulty of the problem. For formulation *PS*, the average values of the gap are 0.8% and 0.2% for the D- and U-instances, respectively. For formulation *AS*, they again depend on the selection of the parameter  $\beta$  and exhibit the same behavior as for the percentage of optimally solved instances and the computing times. In general, the average gaps increase as the value of  $\beta$  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 formulations *PS* and *AS* for every value of  $\beta$  and  $Q$ . For instances of “I30”, optimality becomes more difficult to achieve for larger values of  $\beta$  and smaller values of  $Q$ . For the “I50” instances, optimality is almost unreachable already for values of  $Q < |I|$ . In general, the smaller the value of  $Q$  and the larger value of  $\beta$ , the more difficult to solve the problem.

In general, D-instances are more difficult to solve than U-instances. This suggests that the difficulty of the instances increases as the variety of service intervals ( $s_i$ ) decreases.

### 5.2.1. Impact of the parameter $\beta$

As observed before, for formulation *AS*, parameter  $\beta$  has a crucial impact on the solutions obtained. To better perceive the effect of this parameter, Figure 3 displays the frequency chart that shows for which value of  $\beta$  we first obtain a solution for an instance that uses earliness. 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 ahead of time visits. The average value for this “earliness breakpoint” is  $\bar{\beta}_{bp} = 0.88$  for all best solutions found, and  $\bar{\beta}_{bp} = 0.93$  and  $\bar{\beta}_{bp} = 0.84$  for optimal solutions, for the D- and U-instances, respectively.

Figure 4 shows the number of operators used over all periods for the sets of instances in which all best-found solutions were optimal. For formulation *AS*, the values are displayed for each value of  $\beta$ . This figure shows a clear decrease in the total number of operators used over all periods as the parameter  $\beta$  increases. Figure 5 highlights the trade off between a reduction in the total number of operators and an increase in the total number of service visits. Finally, Figure 6 displays the average

values of  $Time$  for the best solutions found with formulations  $PS$  and  $AS$ . For formulation  $AS$ , the average computing times are depicted for every value of  $\beta$ . The figure supports the hardness of the problem, especially for higher values of  $\beta$  (except for  $\beta = 1.0$ ).

Under the evidence of the reduction in the number of operators used over all periods by applying the aperiodic service policy and the difficulty regarding the value of  $Q$ , the challenge is to find an efficient way to solve the P-MSSP and the A-MSSP for all possible values of  $\beta$  and  $Q$ .

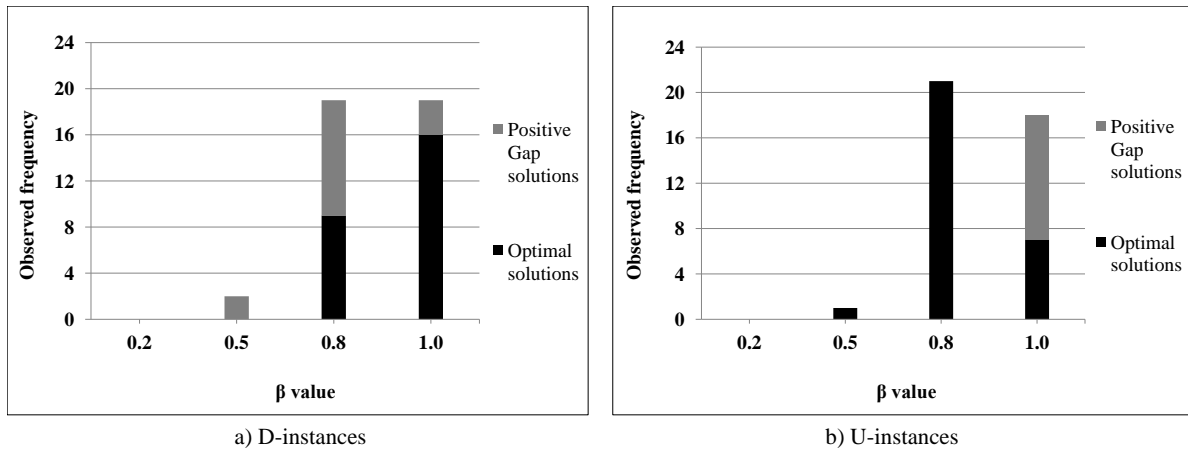


Figure 3: Observed *earliness breakpoint* for formulation  $AS$ .

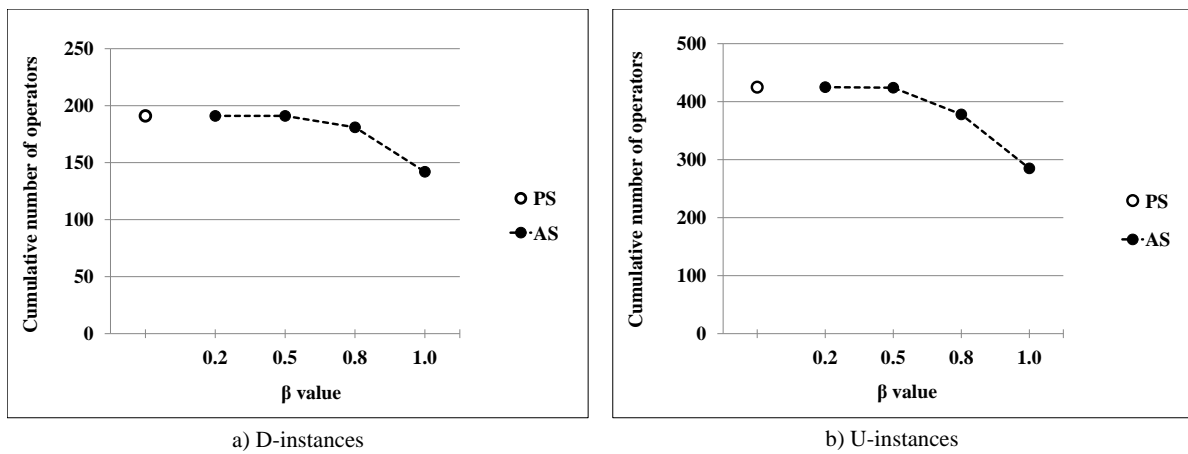


Figure 4: Cumulative number of operators for optimally solved instances.

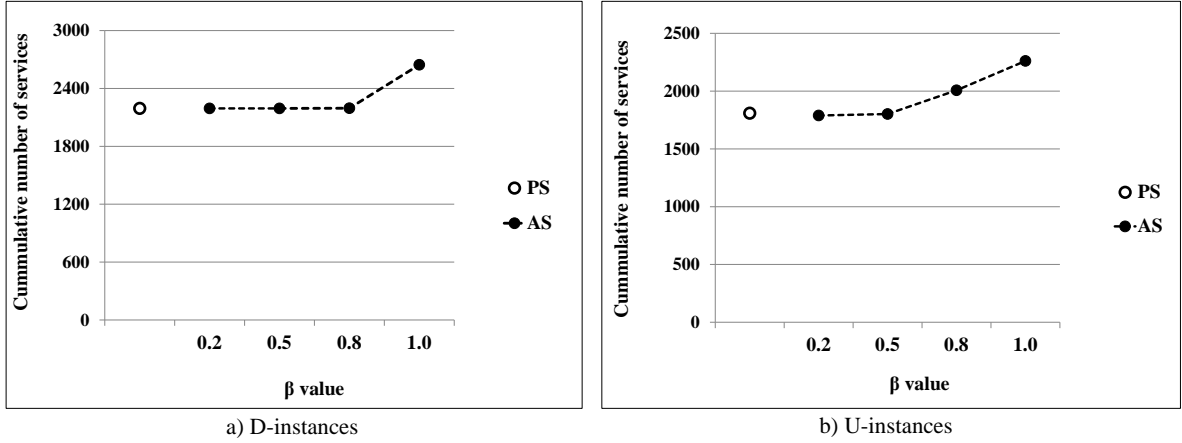


Figure 5: Cumulative number of service visits for optimally solved instances.

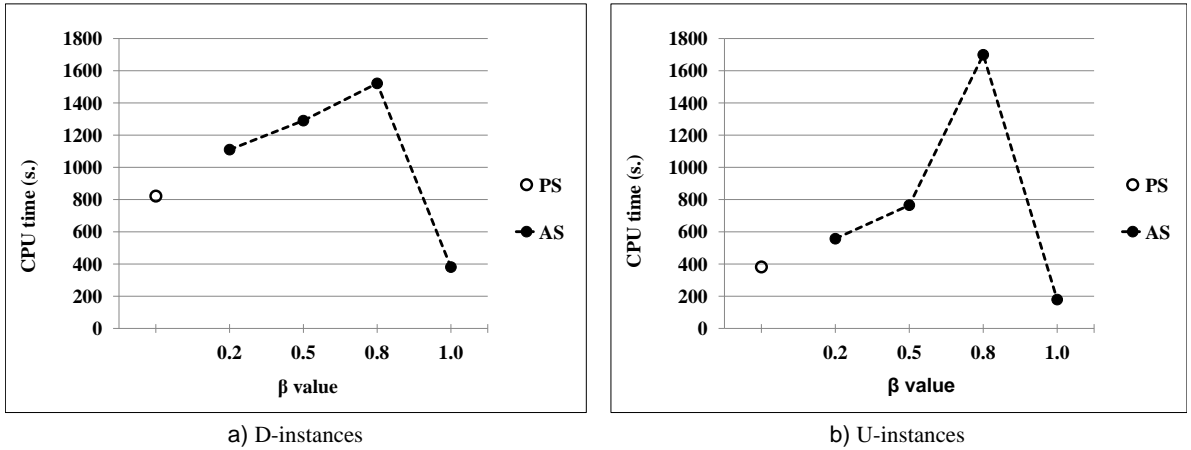


Figure 6: Average CPU times for best-found solutions.

## 6. A Heuristic Algorithm for the MSSP

As we have seen in the previous section, a general purpose solver is only able to solve small and medium instances for formulation *PS* and small instances for formulation *AS* optimally within a reasonable amount of time. Therefore, we next present a heuristic solution algorithm for the P-MSSP and the A-MSSP. The heuristic first constructs an initial solution using a greedy procedure. Afterwards, a local search procedure 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. All phases are described below.

### 6.1. Greedy Procedure

In the greedy procedure we successively build a solution by selecting in each iteration first a calendar-less customer and then finding the best calendar for him. Customers are selected by non-decreasing values of their service intervals  $s_i$ . 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 receive service 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  $Q_{k_t^{max}}$ .  $Q_{k_t^{max}} = Q$  indicates that an additional operator is needed if one more customer should receive service 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 a calendar-less customer  $i_p$  and then determine the best calendar  $C_{i_p} \subseteq T$  for him (in the next section, we show how to do that efficiently). 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 1 gives an outline of the greedy procedure. Without loss of generality, we assume that the index set of customers is sorted by non increasing values of their service intervals, i.e.,  $I = \{i_1, \dots, i_{|I|}\}$  with  $s_{i_1} \leq s_{i_2} \leq \dots \leq s_{i_{|I|}}$ .

---

**Algorithm 1:** Greedy Procedure
 

---

**Data:**  $T, Q, K, I = \{i_1, i_2, \dots, i_{|I|}\}, s_i$ , with  $s_{i_k} \leq s_{i_{k+1}}$ .

- 1 set  $Q_t = k_t^{max} = Q_{k_t^{max}} = 0, t \in T$ ;
  - 2 set  $S = E = \emptyset$ ;
  - 3 **for**  $i_p \in I$  **do**
  - 4 determine the best calendar  $C_{i_p}$  for customer  $i_p$ ;
  - 5 set  $Q_t = Q_t + 1, t \in C_{i_p}$ ;
  - 6 set  $k_t^{max} = \left\lceil \frac{Q_t}{Q} \right\rceil, t \in C_{i_p}$ ;
  - 7 set  $Q_{k_t^{max}} = Q_t - Q \cdot (k_t^{max} - 1), t \in C_{i_p}$ ;
  - 8 set  $S = (S \cup \{t \in C_{i_p} : Q_{k_t^{max}} = Q\}) \setminus (S \cap C_{i_p})$ ;
  - 9 set  $E = E \cup \{t \in C_{i_p} : t \in S, k_t^{max} = |K|\}$ ;
  - 10 **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 periodic service policy

Next, we formulate the search for the best calendar for a selected customer  $i = i_p$  in iteration  $p$  as a shortest path problem. 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  $t \in \{1, \dots, s_i\} \setminus E$ ; (b)  $(v_t, v_{t'})$  with  $t' = t + s_i, t', t \in T \setminus E$ ; and, (c)  $(v_t, v_{|T|+1})$  with  $t \in T \setminus E, t > |T| - s_i$ . Figure 7 depicts 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):

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

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.

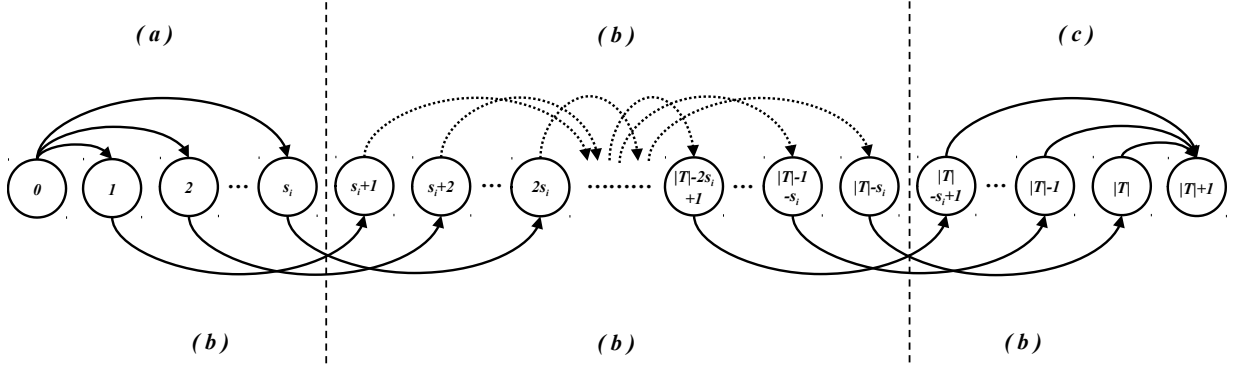


Figure 7: Shortest path network representation for customer  $i \in I$  under the PS policy.

### 6.1.2. Auxiliary shortest path problem for the aperiodic service policy

The set of nodes of the auxiliary network for the aperiodic service policy is identical to the one for the periodic policy. Concerning the arcs,  $A$  now contains three types of arcs: (a)  $(v_0, v_t)$ , with  $t \in \{1, \dots, s_i\} \setminus E$ ; (b)  $(v_t, v_{t'})$  with  $t, t' \in T \setminus E$  where  $t' = t + h$ ,  $h \in \{1, \dots, s_i\}$ ; and, (c)  $(v_t, v_{|T|+1})$  with  $t \in T \setminus E$ ,  $t > |T| - s_i$ . Figure 8 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 customer  $i$ . Thus, we define the following costs associated with the arcs of  $A$  for type (a):

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

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  $s_i - (t' - t)$ . Thus,

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

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  $\beta \frac{Q_k^{max}}{|T|Q}$ . 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.

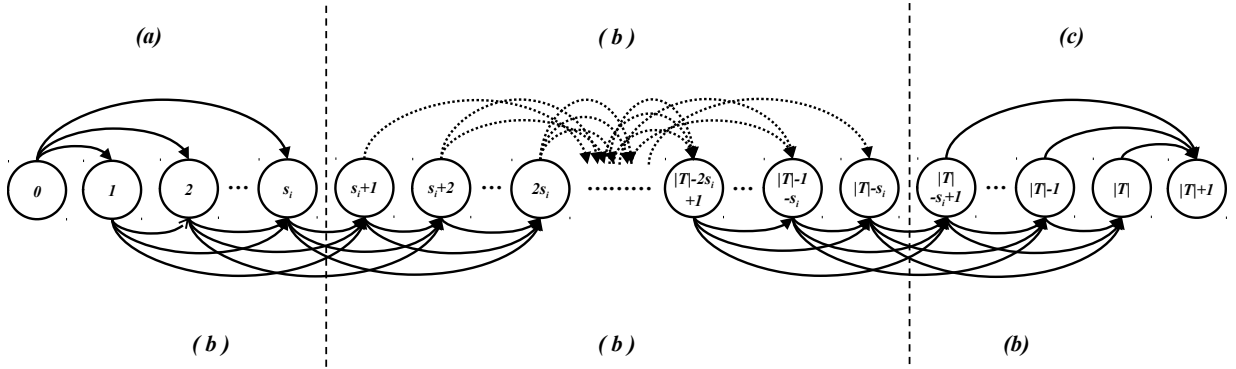


Figure 8: Shortest path network representation for customer  $i \in I$  under the AS policy.

## 6.2. Local Search Procedure

In the local search procedure we try to improve the solution obtained by the greedy procedure. To that end, two different neighborhoods 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, decrease 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.



### 6.2.1. Neighborhoods

The *single-customer neighborhood*, 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 calendar  $C_i$  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  $t > |T| - s_i$  or  $r > |T| - s_j$ . 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  $t' \in \{1, \dots, s_i\}$ ,  $r' \in \{1, \dots, s_j\}$ ;  $(A_2)$  both nodes are of type  $(V_2)$  with  $t' = t + s_i$  and  $r' = r + s_j$ ; 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 calendars for customers  $i$  and  $j$ .

Unfortunately, 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 consider the two-customers neighborhood only for the PC policy.

### 6.2.2. Search strategies

The neighborhoods described above are used within several strategies which differ from each other on how the customers are selected. The strategies that explore neighborhood  $N_1$  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 served by this operator. If there exists more than one customer, they are selected by increasing index.

The strategy that explores neighborhood  $N_2$  is the following:

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

The local search procedure uses two distinct combinations of strategies:

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

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

### 6.2.3. Diversification by Perturbation

To diversify the search, we apply the local search not only to the solution produced by the greedy procedure but as well to a set of perturbed solutions. We obtain the latter with 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  $|S| \sim U(2, |I|)$ . 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.

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

Algorithm 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 algorithm is applied and then local search followed by the destroy & repair procedure are applied for  $MaxDiv$  iterations.

---

**Algorithm 2:** Heuristic Algorithm

---

**Data:**  $MaxDiv$

```
1  $GreedySol \leftarrow Greedy$ ;  
2  $BestSol \leftarrow GreedySol$ ;  
3  $sol \leftarrow GreedySol$ ;  
4  $iter \leftarrow 0$  ;  
5 while  $iter \leq MaxDiv$  do  
6    $LocalSol \leftarrow LocalSearch(sol)$ ;  
7   if  $F(LocalSol) \leq F(BestSol)$  then  
8      $BestSol \leftarrow LocalSol$ ;  
9   end  
10   $sol \leftarrow Destroy\&Repair(GreedySol)$ ;  
11   $iter \leftarrow iter + 1$  ;  
12 end
```

---

## 7. Computational Experience with the Heuristic

To assess the effectiveness of the heuristic algorithm described above, we ran a second series of computational experiments, using the same benchmark instances as in Section 5. We tried all search strategies described in Section 6.2.2 and chose the best one for the PS and, respectively, the AS policy. The algorithm was coded in C++. Recall that we randomly generated five problem instances for each combination of values for  $|I|$  and  $Q$ .

For each  $|I|$  and  $Q$ , Table 4 shows the summary of the results of the best strategy for the PS and AS policy for the D- and U-instances. The detailed results can be found in Tables C.1 and C.2 in Appendix C. Columns  $z$  and  $e$  display the total number of operators used and the total earliness, respectively, over the five instances. Columns  $HGap$  denote the relative difference between the best solution found by CPLEX 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  $Time$  to display the average computing time (in seconds) needed by the heuristic.

PS		AS															
		$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$			
<i>z</i>	<i>HGap</i>	<i>Time</i>		<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>		
D-instances																	
<i>I10</i>	<i>Q5</i>	51	0.00	1.9	51	0	0.00	3.0	51	0	0.00	3.0	46	6	0.00	2.5	
	<i>Q10</i>	40	0.00	2.9	40	0	0.00	2.9	40	0	0.00	3.0	37	8	0.00	4.8	
	<i>Q5</i>	140	0.00	9.1	140	0	-0.65	28.7	140	0	0.00	66.4	137	8	1.59	221.5	
<i>I30</i>	<i>Q10</i>	79	0.00	15.7	79	0	0.00	20.7	79	0	0.00	40.2	78	5	2.32	43.7	
	<i>Q30</i>	50	0.00	13.7	50	0	0.00	24.3	50	0	0.00	29.1	50	0	0.00	36.5	
	<i>Q5</i>	242	-0.44	30.3	242	0	-0.44	28.3	242	0	-0.44	53.2	238	5	-0.10	58.2	
<i>I50</i>	<i>Q10</i>	132	0.00	56.4	132	0	0.00	57.0	131	0	0.00	56.6	128	9	2.08	89.1	
	<i>Q50</i>	50	0.00	58.7	50	0	0.00	23.7	50	0	0.00	72.9	50	0	0.00	65.9	
U-instances																	
<i>I10</i>	<i>Q5</i>	55	0.00	1.8	55	0	0.00	3.2	55	0	0.00	3.5	49	8	0.00	3.0	
	<i>Q10</i>	55	0.00	3.8	55	0	0.00	3.6	55	0	0.00	3.9	47	8	0.00	5.8	
	<i>Q5</i>	109	0.00	9.5	109	0	0.00	47.7	109	0	0.00	58.3	109	8	3.06	58.7	
<i>I30</i>	<i>Q10</i>	71	0.00	42.5	71	0	0.00	20.2	71	0	0.00	66.5	67	8	2.49	33.9	
	<i>Q30</i>	70	0.00	26.8	70	0	0.00	23.5	70	0	0.00	56.8	65	9	2.14	42.8	
	<i>Q5</i>	185	-0.51	48.7	185	0	-0.51	108.6	185	0	0.00	82.8	184	1	-0.54	9.4	
<i>I50</i>	<i>Q10</i>	99	0.00	75.7	99	0	0.00	68.4	99	0	0.00	65.7	99	5	1.79	152.1	
	<i>Q50</i>	75	0.00	56.3	75	0	0.00	41.6	75	0	0.00	219.4	72	12	2.37	219.2	

Table 4: Average heuristic results for the D- and U-instances.

For the PS policy, the best results are obtained after applying the local search procedure using the search strategy  $S_{AC}$  described in Section 6.2.2 over 100 perturbed solutions. The heuristic obtains optimal solutions for all instances in which CPLEX does. For instances not optimally solved by CPLEX, the heuristic produces the same (or a better) upper bound with considerable less time (see Table 2 for comparison). On average, the heuristic solutions are 0.06% better than those of CPLEX.

For the AS policy, best results have been obtained using the search strategy  $S_{AB}$  (see Section 6.2.2) over 100 perturbed solutions. Table 4 displays the results for each value of  $\beta$  and for each series of instances. On average, the heuristic solutions are 0.14% and 0.34% worse than those of 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  $\beta$ , for  $\beta = 1.0$  the heuristic finds the same solutions as CPLEX, 92.5% and 100% of which were proven to be optimal, for the D- and U-instances, respectively. The same results are obtained for the U-instances and  $\beta = 0.5$ , in which 95% of the instances were proven to be optimal. For the D-instances with  $\beta \in \{0.2, 0.5\}$  and the U-instances with  $\beta = 0.2$ , the heuristic algorithm outperforms CPLEX on average because we have negative  $HGaps$ . The worst outcome of the heuristic is observed for  $\beta = 0.8$ , with average  $HGaps$  of 0.74 and 1.41 for the D- and

U-instances, respectively. Observe however that for instances “I30”, the heuristic finds the same solutions as CPLEX for all instances that were solved optimally. Moreover, for instances “I50”, the heuristic outperforms CPLEX for 13.33% and 26.67% of the cases for the U- and D-instances, respectively.

Regarding CPU times, it can be observed that the heuristic CPU times are shorter than those of CPLEX. Excluding the instances with trivial capacities, i.e.,  $Q = |I|$ , the average computing times for the heuristic are significantly smaller than those of CPLEX. For the rest of the instances, the differences in CPU times are not significant.

Finally, for each  $I$  and  $Q$ , Table 5 summarizes for the PS and AS policy the contribution of each ingredient of the heuristic to its overall performance over each group of D- and U-instances. The detailed results can be found in Tables C.3 and C.4 in Appendix C. Columns  $HG_G$ ,  $HG_{LS}$ , and  $HG_P$  denote the averages over the five instances in each group of the relative difference between the best solution found by CPLEX and the value of the best solution found by the greedy procedure, the local search applied to the solution of the greedy procedure, and at the end of the heuristic when the local search has been applied over the 100 perturbed solutions. As can be seen the local search produces a considerable improvement over the greedy solution, particularly for  $\beta = 0.8$  when the greedy solution is usually not very good. However, the repetition over the perturbed solutions is significant for closing the gaps.

## 8. Conclusions

We have presented a novel problem for multi-period service scheduling. The objective is to reduce the total number of operators while ensuring that two consecutive visits to a customer do not exceed its service interval. Two different service policies have been addressed. While the PS policy imposes a visiting activity with a fixed time interval, the AS policy allows visits before the end of the service interval. Two mathematical formulations have been presented, one for each service policy. The one for AS policy introduces a second term in the objective function to penalize ahead of time visits. This allows to model several scenarios, depending on the penalty weight. Computational results indicate that allowing for ahead of time visits reduces the number of operators for an appropriate trade-off of the objective function terms. Moreover, we have developed a heuristic algorithm that obtains good solutions within a noticeable short time.

		PS			AS											
					$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$		
		$HG_G$	$HG_{LS}$	$HG_P$	$HG_G$	$HG_{LS}$	$HG_P$	$HG_G$	$HG_{LS}$	$HG_P$	$HG_G$	$HG_{LS}$	$HG_P$	$HG_G$	$HG_{LS}$	$HG_P$
D-instances																
I10	Q5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.05	0.95	0.00	5.00	2.50	0.00
	Q10	6.00	0.00	0.00	2.00	0.00	0.00	6.00	0.00	0.00	6.71	0.00	0.00	0.00	0.00	0.00
I30	Q5	3.43	0.67	0.00	2.05	0.00	-0.65	3.43	0.00	0.00	14.26	1.59	1.59	4.64	3.90	0.00
	Q10	8.99	2.61	0.00	6.38	0.00	0.00	8.99	1.18	0.00	37.70	2.32	2.32	5.76	1.43	0.00
	Q30	10.00	0.00	0.00	6.00	0.00	0.00	10.00	0.00	0.00	48.50	12.00	0.00	0.00	0.00	0.00
I50	Q5	2.01	0.80	-0.44	1.21	0.00	-0.44	2.01	0.00	-0.44	5.39	0.11	-0.10	2.13	2.13	0.00
	Q10	6.03	2.23	0.00	3.80	0.00	0.00	6.82	2.20	0.00	39.33	2.08	2.08	5.88	5.88	0.00
	Q50	10.00	0.00	0.00	2.00	0.00	0.00	10.00	0.00	0.00	71.50	29.50	0.00	0.00	0.00	0.00
U-instances																
I10	Q5	7.60	0.00	0.00	5.93	0.00	0.00	7.60	0.00	0.00	18.08	0.00	0.00	2.50	2.50	0.00
	Q10	7.60	0.00	0.00	5.93	0.00	0.00	7.60	0.00	0.00	12.18	0.00	0.00	0.00	0.00	0.00
I30	Q5	7.56	2.89	0.00	9.02	2.89	0.00	8.39	4.63	0.00	38.77	3.06	3.06	3.95	3.04	0.00
	Q10	10.10	1.67	0.00	8.43	1.67	0.00	10.10	4.43	0.00	62.89	7.68	2.49	14.90	5.30	0.00
	Q30	5.87	0.00	0.00	5.87	0.00	0.00	5.87	0.00	0.00	37.64	10.23	2.14	0.00	0.00	0.00
I50	Q5	5.90	1.59	-0.51	7.63	1.59	-0.51	8.06	5.40	0.00	32.31	1.18	-0.54	2.77	2.77	0.00
	Q10	13.24	6.06	0.00	11.17	5.22	0.00	13.24	10.13	0.00	108.53	3.85	1.79	8.84	5.50	0.00
	Q50	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	83.19	11.19	2.37	0.00	0.00	0.00

Table 5: Average relative contribution of each heuristic phase.

## Acknowledgements

This research has been partially supported by the Spanish Ministry of Economy and Competitiveness through grant MTM2012-36-163-C06-05 and EDRF funds, and by grant NI 521/6-1 of the German Research Foundation (DFG). The research of the first author has been partially through of the Mexican National Council for Science and Technology (CONACyT) through grant 310857. This support is gratefully acknowledged.

## References

- [1] Anily, S., Glass, C., Hassin, R., 1998. The scheduling of maintenance service. *Discrete Applied Mathematics* 82 (1-3), 27–42.
- [2] Bar-Noy, A., Bhatia, R., Naor, J., Schieber, B., 2002. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research* 27 (3), 518–544.
- [3] Bar-Noy, A., Ladner, R., 2003. Windows scheduling problems for broadcast systems. *SIAM Journal on Computing* 32 (4), 1091–1113.
- [4] Bar-Noy, A., Ladner, R., Tamir, T., 2007. Windows scheduling as a restricted version of bin packing. *ACM Transactions on Algorithms* 3 (3).
- [5] Bar-Noy, A., Ladner, R., Tamir, T., Van De Grift, T., 2012. Windows scheduling of arbitrary-length jobs on multiple machines. *Journal of Scheduling* 15 (2), 141–155.
- [6] Bertazzi, L., 2008. Analysis of direct shipping policies in an inventory-routing problem with discrete shipping times. *Management Science* 54 (4), 748–762.

- [7] Campbell, A., Hardin, J., 2005. Vehicle minimization for periodic deliveries. *European Journal of Operational Research* 165, 668–684.
- [8] Campbell, A. M., Wilson, J. H., 2014. Forty years of periodic vehicle routing. *Networks* 63 (1), 2–15.  
URL <http://dx.doi.org/10.1002/net.21527>
- [9] Coelho, L., Cordeau, J., Laporte, G., 2013. Thirty years of inventory routing. *Transportation Science* 48 (1), 1–19.
- [10] Corominas, A., Kubiak, W., Pastor, R., 2010. Mathematical programming modeling of the response time variability problem. *European Journal of Operational Research* 200 (2), 347–357.
- [11] Corominas, A., Kubiak, W., Pastor, R., 2012. Heuristics for the response time variability problem. *European Journal of Industrial Engineering* 6 (6), 751–764.
- [12] Fernández, E., Kalcsics, J., Nickel, S., Ríos-Mercado, R., 2010. A novel territory design model arising in the implementation of the WEEE-directive. *Journal of the Operation Research Society* 61 (3), 503–514.
- [13] Gallego, G., Simchi-Levi, D., 1990. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer r-systems. *Management Science* 36 (2), 240–243.
- [14] García-Villoria, A., Corominas, A., Delorme, X., Dolgui, A., Kubiak, W., Pastor, R., 2013. A branch and bound algorithm for the response time variability problem. *Journal of Scheduling* 16 (2), 243–252.
- [15] Grigoriev, A., van de Klundert, J., Spieksma, F., 2006. Modeling and solving the periodic maintenance problem. *European Journal of Operational Research* 172 (3), 783–797.
- [16] Han, C.-C., Lin, K.-J., Hou, C.-J., 1996. Distance-constrained scheduling and its applications to real-time systems. *IEEE Transactions on Computers* 45 (7), 814–826.
- [17] Herrmann, J., 2011. Using aggregation to reduce response time variability in cyclic fair sequences. *Journal of Scheduling* 14 (1), 39–55.
- [18] Jeffries, T., O’Hanley, J., 2012. Optimal location and scheduling of blood collection facilities. In: *OR54, University of Edinburgh*, 4-6 September 2012.
- [19] Korst, J., Aarts, E., Lenstra, J. K., Wessels, J., 1994. Periodic assignment and graph colouring. *Discrete Applied Mathematics* 51 (3), 291–305.
- [20] McClurg, T., Chand, S., 2002. A parallel machine replacement model. *Naval Research Logistics* 49 (3), 275–287.
- [21] Queiruga, D., Walther, G., González-Benito, J., Spengler, T., 2008. Evaluation of sites for the location of WEEE recycling plants in Spain. *Waste Management* 28 (1), 181 – 190.
- [22] Sbihi, A., Eglese, R., 2007. Combinatorial optimization and green logistics. *4OR: A Quarterly Journal of Operational Research* 5 (2), 99–116.
- [23] Xu, Y., Yu, H., Liu, K., Tang, L., 2011. Inventory replenishment scheduling to minimize the number of vehicles. pp. 74–78.
- [24] Yavuz, M., Jeffcoat, D., 2007. An analysis and solution of the sensor scheduling problem. *Lecture Notes in Control and Information Sciences* 369, 167–177.

## Appendix A. Proof of propositions of Section 4.3

**Proposition 1.**  $\Omega_0 = \Omega_1$ .

*Proof.* (a)  $\Omega_1 \subseteq \Omega_0$ . Let us suppose there exists  $(x, y, z, f) \in \Omega_1$  such that  $(x, y, z, f) \notin \Omega_0$ . Then, there exist indices  $i \in I$  and  $t \in \{0, \dots, |T| - s_i\}$  with  $x_i^t > \sum_{h=1}^{s_i} x_i^{t+h}$ . By definition,  $x_i^t = 1$  and  $x_i^{t+h} = 0$  for all  $h \in \{1, \dots, s_i\}$ . By (11),  $\sum_{h=1}^{s_i} f_i^{th} = 1$ , so  $f_i^{th'} = 1$  for some  $h' \in \{1, \dots, s_i\}$ . But then,

$$0 = x_i^{t+h'} = \sum_{h=\max\{1, t+h'-(|T|-s_i)\}}^{\min\{s_i, t+h'\}} f_i^{t+h'-h, h} \geq f_i^{th'} = 1,$$

as  $h' \in [\max\{1, t+h'-(|T|-s_i)\}, \min\{s_i, t+h'\}]$ .

(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,  $\sum_{h=\max\{1, t'-(|T|-s_i)\}}^{\min\{s_i, t'\}} f_i^{t'-h, h} \neq x_i^{t'}$  for some  $t' \in T$ .

We first consider the case that  $\sum_{h=\max\{1, t'-(|T|-s_i)\}}^{\min\{s_i, t'\}} f_i^{t'-h, h} < x_i^{t'}$ . By definition,  $x_i^{t'} = 1$  and  $f_i^{t'-h, h} = 0$  for  $h = \max\{1, t'-(|T|-s_i)\}, \dots, \min\{s_i, t'\}$ . If  $t' \geq 1$  is the first service period for customer  $i$ , then  $t' \leq s_i$ . Setting  $t = 0$  and  $h = t'$  in (12), we get

$$2 = x_i^0 + x_i^{t'} \leq \sum_{s=1}^{t'-1} x_i^s + f_i^{0, t'} + 1 = 1,$$

as  $t' \in \{\max\{1, t'-(|T|-s_i)\}, \dots, \min\{s_i, t'\}\}$ . If  $t'$  is not the first service period for  $i$ , then let  $h' \geq 1$  be minimal with  $x_i^{t'-h'} = 1$ , i.e.,  $t' - h' \geq 1$  is the service period immediately preceding  $t'$ . By (10),  $h' \leq s_i$ . Thus, setting  $t = t' - h'$  and  $h = h'$  in (12), we again obtain a contradiction.

Next, we consider the case that  $\sum_{h=\max\{1, t'-(|T|-s_i)\}}^{\min\{s_i, t'\}} f_i^{t'-h, h} > x_i^{t'}$ . First, we assume that  $x_i^{t'} = 0$ . Then, there exists at least one  $h' \in \{\max\{1, t'-(|T|-s_i)\}, \dots, \min\{s_i, t'\}\}$  with  $f_i^{t'-h', h'} = 1$ . From (11) we get  $x_i^{t'-h'} = 1$  and, subsequently, from (10) that there exists  $\tilde{h}$  with  $x_i^{t'-h'+\tilde{h}} = 1$  ( $t' - h' \leq |T| - s_i$  by definition of  $h'$ ). Without loss of generality, assume that  $1 \leq \tilde{h} \leq s_i$  is minimal with respect to this property. By assumption,  $h' \neq \tilde{h}$ . Setting  $t = t' - h' \geq 1$  and  $h = \tilde{h}$  in (12), we must have  $f_i^{t'-h', \tilde{h}} = 1$ . But this contradicts (11) as  $h' \neq \tilde{h}$ . Next, we assume that  $x_i^{t'} = 1$ . Then, there exist  $h^1, h^2 \in \{\max\{1, t'-(|T|-s_i)\}, \dots, \min\{s_i, t'\}\}$  with  $h^1 \neq h^2$  and  $f_i^{t'-h^1, h^1} = f_i^{t'-h^2, h^2} = 1$ . Without loss of generality, let  $h^1 > h^2$ . If there are more than two such



values, we pick two successive ones. From (11) we get  $x_i^{t'-h^1} = x_i^{t'-h^2} = 1$  as  $t'-h^1, t'-h^2 \leq |T| - s_i$ . Setting  $t = t' - h^1$  and  $h = h^1 - h^2$  in (12), we must have  $f_i^{t'-h^1, h^1-h^2} = 1$ . But this contradicts (11) as  $h^2 > 0$ .  $\square$

**Proposition 2.**  $\bar{\Omega}_1 \subseteq \bar{\Omega}_0$ .

*Proof.* Consider  $(x, y, z, f) \in \bar{\Omega}_1$  and let  $i \in I$ . From (20) follows that  $f_i^{t-h, h} \leq x_i^t$  for all  $t \in T$  and  $h \in \{\max\{1, t - (|T| - s_i)\}, \dots, \min\{s_i, t\}\}$ . This is equivalent to  $f_i^{th} \leq x_i^{t+h}$  for all  $t \in \{0, \dots, |T| - s_i\}$  and  $h \in \{1, \dots, s_i\}$ . Thus, for any  $t \in \{0, \dots, |T| - s_i\}$ , we have  $\sum_{h=1}^{s_i} f_i^{th} \leq \sum_{h=1}^{s_i} x_i^{t+h}$ . In addition, by (11),  $x_i^t = \sum_{h=1}^{s_i} f_i^{th}$ . Thus,  $x_i^t \leq \sum_{h=1}^{s_i} x_i^{t+h}$ , which is in fact (10) and therefore  $(x, y, z, f) \in \bar{\Omega}_0$ .  $\square$

## Appendix B. Detailed results for the MSSP formulations

Instance	PS			AS																
				$\beta = 0.2$				$\beta = 0.5$				$\beta = 0.8$				$\beta = 1.0$				
	<i>z</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	
I10	<i>D_I10_Q5_C1</i>	6	0.0	0.5	6	0	0.0	0.2	6	0	0.0	0.3	6	0	0.0	0.3	6	5	0.0	0.3
	<i>D_I10_Q5_C2</i>	12	0.0	1.0	12	0	0.0	0.9	12	0	0.0	2.0	8	10	0.0	4.3	8	12	0.0	2.8
	<i>D_I10_Q5_C3</i>	9	0.0	1.1	9	0	0.0	1.1	9	0	0.0	1.5	8	1	0.0	5.3	8	60	0.0	42.9
	<i>D_I10_Q5_C4</i>	12	0.0	0.9	12	0	0.0	1.0	12	0	0.0	1.8	11	1	0.0	5.4	10	36	0.0	4.6
	<i>D_I10_Q5_C5</i>	12	0.0	0.7	12	0	0.0	0.5	12	0	0.0	1.0	11	2	0.0	1.7	11	13	0.0	0.8
	<i>D_I10_Q10_C1</i>	5	0.0	0.2	5	0	0.0	0.3	5	0	0.0	0.4	5	0	0.0	0.7	4	48	0.0	0.6
	<i>D_I10_Q10_C2</i>	10	0.0	0.1	10	0	0.0	0.3	10	0	0.0	0.5	9	2	0.0	1.9	7	77	0.0	2.1
	<i>D_I10_Q10_C3</i>	5	0.0	0.2	5	0	0.0	0.3	5	0	0.0	0.3	4	3	0.0	0.5	4	55	0.0	0.3
	<i>D_I10_Q10_C4</i>	10	0.0	0.1	10	0	0.0	0.4	10	0	0.0	0.5	9	3	0.0	1.5	7	86	0.0	0.5
	<i>D_I10_Q10_C5</i>	10	0.0	0.1	10	0	0.0	0.3	10	0	0.0	0.4	10	0	0.0	2.0	7	76	0.0	0.5
I30	<i>D_I30_Q5_C1</i>	23	0.0	817.1	23	0	0.0	1558.8	23	0	0.0	1676.2	23	0	7.0	3600.8	22	87	0.0	148.4
	<i>D_I30_Q5_C2</i>	29	0.0	1550.4	29	0	4.1	3601.7	29	0	7.2	3601.5	27	6	7.4	3602.3	27	41	0.0	14.9
	<i>D_I30_Q5_C3</i>	28	0.0	1171.8	28	0	0.0	2380.1	28	0	7.1	3602.5	26	4	5.2	3602.6	26	84	0.0	220.3
	<i>D_I30_Q5_C4</i>	30	0.0	2227.8	31	0	9.0	3601.4	29	1	9.3	3602.9	28	4	6.9	3606.7	27	74	0.0	94.7
	<i>D_I30_Q5_C5</i>	30	0.0	2070.9	30	0	0.0	1920.2	30	0	8.7	3602.5	28	5	6.3	3604.7	28	79	0.0	6.8
	<i>D_I30_Q10_C1</i>	14	0.0	4.4	14	0	0.0	12.5	14	0	0.0	30.7	14	0	0.0	578.1	12	94	0.0	1668.7
	<i>D_I30_Q10_C2</i>	17	0.0	35.7	17	0	0.0	65.7	17	0	0.0	275.3	15	6	0.0	1624.6	14	72	0.0	355.6
	<i>D_I30_Q10_C3</i>	15	0.0	5.8	15	0	0.0	25.7	15	0	0.0	83.4	15	0	0.0	492.8	14	95	8.3	3605.2
	<i>D_I30_Q10_C4</i>	18	0.0	47.2	18	0	0.0	176.2	18	0	0.0	737.9	16	4	11.8	3603.2	15	111	7.6	3600.1
	<i>D_I30_Q10_C5</i>	15	0.0	2.6	15	0	0.0	7.4	15	0	0.0	32.7	15	0	0.0	370.4	15	83	8.1	3602.2
	<i>D_I30_Q30_C1</i>	10	0.0	0.3	10	0	0.0	1.1	10	0	0.0	3.0	10	0	0.0	9.4	7	322	0.0	5.7
	<i>D_I30_Q30_C2</i>	10	0.0	0.1	10	0	0.0	0.8	10	0	0.0	1.6	10	0	0.0	10.1	7	364	0.0	3.7
	<i>D_I30_Q30_C3</i>	10	0.0	0.1	10	0	0.0	0.8	10	0	0.0	2.8	10	0	0.0	7.2	7	261	0.0	4.9
	<i>D_I30_Q30_C4</i>	10	0.0	0.1	10	0	0.0	0.9	10	0	0.0	1.4	10	0	0.0	7.4	7	252	0.0	3.1
	<i>D_I30_Q30_C5</i>	10	0.0	0.1	10	0	0.0	0.9	10	0	0.0	2.3	10	0	0.0	8.3	7	236	0.0	6.3
I50	<i>D_I50_Q5_C1</i>	45	5.8	3601.7	45	0	5.8	3601.7	45	0	5.8	3600.6	43	3	3.1	3601.2	43	92	0.0	97.7
	<i>D_I50_Q5_C2</i>	47	4.3	3602.3	47	0	4.3	3602.2	47	0	4.3	3604.2	46	3	3.7	3601.2	45	94	0.0	653.4
	<i>D_I50_Q5_C3</i>	49	4.1	3602.0	49	0	4.1	3602.2	49	0	4.1	3603.7	48	3	3.6	3602.6	47	173	0.0	93.8
	<i>D_I50_Q5_C4</i>	50	5.2	3601.9	50	0	5.2	3600.3	50	0	5.2	3603.9	48	4	3.3	3604.1	48	105	0.0	113.2
	<i>D_I50_Q5_C5</i>	52	2.7	3601.8	52	0	2.7	3602.3	52	0	2.7	3603.0	51	1	1.3	3601.0	51	106	0.0	116.1
	<i>D_I50_Q10_C1</i>	25	0.0	1604.6	25	0	7.6	3601.0	25	0	15.2	3601.7	24	1	12.6	3601.1	22	117	0.0	51.1
	<i>D_I50_Q10_C2</i>	24	0.0	433.6	24	0	0.0	2481.6	24	0	0.0	1902.9	24	0	6.3	3603.1	23	182	0.0	47.0
	<i>D_I50_Q10_C3</i>	27	0.0	1048.0	27	0	0.0	467.6	27	0	11.9	3600.1	25	3	6.2	3603.9	24	111	0.0	221.2
	<i>D_I50_Q10_C4</i>	28	10.0	3600.9	28	0	15.0	3601.6	26	1	12.2	3602.9	24	6	7.1	3602.7	24	149	0.0	269.3
	<i>D_I50_Q10_C5</i>	28	0.0	232.5	28	0	0.0	2877.3	28	0	8.2	3602.0	28	0	9.6	3602.2	26	64	0.0	133.0
	<i>D_I50_Q50_C1</i>	10	0.0	0.1	10	0	0.0	1.3	10	0	0.0	3.5	10	0	0.0	13.8	7	633	0.0	12.9
	<i>D_I50_Q50_C2</i>	10	0.0	0.1	10	0	0.0	1.4	10	0	0.0	2.6	10	0	0.0	16.6	7	465	0.0	8.8
	<i>D_I50_Q50_C3</i>	10	0.0	0.1	10	0	0.0	1.3	10	0	0.0	2.6	10	0	0.0	20.9	7	362	0.0	9.2
	<i>D_I50_Q50_C4</i>	10	0.0	0.1	10	0	0.0	2.0	10	0	0.0	2.7	10	0	0.0	14.6	7	391	0.0	9.2
	<i>D_I50_Q50_C5</i>	10	0.0	0.1	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: Detailed CPLEX results for the MSSP formulations *PS* and *AS* (D-instances)<sup>1,2</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 2.

<sup>2</sup>Entries with less than 3600 seconds in the *Time* column indicate that the corresponding values are optimal.

Instance	PS			AS																
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$							
	<i>z</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>Gap</i>	<i>Time</i>					
I10	<i>U</i> <i>I</i> 10 <i>Q</i> 5 <i>C</i> 1	7	0.0	0.5	7	0	0.0	1.3	7	0	0.0	2.6	6	2	0.0	8.1	5	35	0.0	0.3
	<i>U</i> <i>I</i> 10 <i>Q</i> 5 <i>C</i> 2	13	0.0	0.6	13	0	0.0	1.5	13	0	0.0	4.3	9	9	0.0	12.5	8	37	0.0	9.6
	<i>U</i> <i>I</i> 10 <i>Q</i> 5 <i>C</i> 3	12	0.0	0.5	12	0	0.0	1.3	12	0	0.0	1.8	11	2	0.0	33.8	9	38	0.0	8.7
	<i>U</i> <i>I</i> 10 <i>Q</i> 5 <i>C</i> 4	13	0.0	0.5	13	0	0.0	1.3	13	0	0.0	6.2	10	7	0.0	39.0	9	51	0.0	6.6
	<i>U</i> <i>I</i> 10 <i>Q</i> 5 <i>C</i> 5	10	0.0	0.5	10	0	0.0	0.9	10	0	0.0	1.9	10	0	0.0	8.1	8	36	0.0	2.1
	<i>U</i> <i>I</i> 10 <i>Q</i> 10 <i>C</i> 1	7	0.0	0.3	7	0	0.0	0.6	7	0	0.0	1.1	5	2	0.0	2.7	3	37	0.0	0.2
	<i>U</i> <i>I</i> 10 <i>Q</i> 10 <i>C</i> 2	13	0.0	0.1	13	0	0.0	0.6	13	0	0.0	2.6	10	5	0.0	9.4	7	113	0.0	1.2
	<i>U</i> <i>I</i> 10 <i>Q</i> 10 <i>C</i> 3	12	0.0	0.3	12	0	0.0	0.6	12	0	0.0	1.3	10	4	0.0	4.3	7	72	0.0	1.1
	<i>U</i> <i>I</i> 10 <i>Q</i> 10 <i>C</i> 4	13	0.0	0.1	13	0	0.0	1.0	13	0	0.0	1.7	9	11	0.0	8.2	7	66	0.0	0.9
	<i>U</i> <i>I</i> 10 <i>Q</i> 10 <i>C</i> 5	10	0.0	0.2	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	<i>U</i> <i>I</i> 30 <i>Q</i> 5 <i>C</i> 1	18	0.0	111.0	18	0	0.0	225.9	18	0	0.0	541.0	18	0	4.1	3600.2	17	63	0.0	5.2
	<i>U</i> <i>I</i> 30 <i>Q</i> 5 <i>C</i> 2	22	0.0	855.6	22	0	0.0	1938.7	22	0	0.0	2098.3	21	1	3.6	3611.2	21	95	0.0	7.3
	<i>U</i> <i>I</i> 30 <i>Q</i> 5 <i>C</i> 3	22	0.0	122.0	22	0	0.0	570.4	22	0	0.0	933.5	22	0	3.6	3603.2	22	118	0.0	24.7
	<i>U</i> <i>I</i> 30 <i>Q</i> 5 <i>C</i> 4	24	0.0	128.4	24	0	0.0	407.9	24	0	0.0	449.2	24	0	3.3	3604.4	24	93	0.0	113.4
	<i>U</i> <i>I</i> 30 <i>Q</i> 5 <i>C</i> 5	23	0.0	404.9	23	0	0.0	251.1	23	0	0.0	1523.5	22	2	3.8	3601.0	22	52	0.0	11.0
	<i>U</i> <i>I</i> 30 <i>Q</i> 10 <i>C</i> 1	12	0.0	6.0	12	0	0.0	28.0	12	0	0.0	79.8	11	2	0.0	861.9	9	78	0.0	120.7
	<i>U</i> <i>I</i> 30 <i>Q</i> 10 <i>C</i> 2	15	0.0	18.0	15	0	0.0	76.8	15	0	0.0	326.3	13	4	0.0	1283.2	11	80	0.0	220.0
	<i>U</i> <i>I</i> 30 <i>Q</i> 10 <i>C</i> 3	14	0.0	11.4	14	0	0.0	37.3	14	0	0.0	284.0	13	2	0.0	1186.4	11	74	0.0	524.6
	<i>U</i> <i>I</i> 30 <i>Q</i> 10 <i>C</i> 4	15	0.0	10.8	15	0	0.0	41.2	15	0	0.0	398.9	13	4	0.0	757.0	12	69	0.0	257.6
	<i>U</i> <i>I</i> 30 <i>Q</i> 10 <i>C</i> 5	15	0.0	10.1	15	0	0.0	67.7	15	0	0.0	214.9	13	5	0.0	1263.7	12	112	0.0	1700.0
	<i>U</i> <i>I</i> 30 <i>Q</i> 30 <i>C</i> 1	12	0.0	0.3	12	0	0.0	4.1	11	1	0.0	54.5	10	4	0.0	254.3	6	511	0.0	6.0
	<i>U</i> <i>I</i> 30 <i>Q</i> 30 <i>C</i> 2	15	0.0	0.2	15	0	0.0	4.9	15	0	0.0	102.7	12	8	0.0	571.0	7	258	0.0	10.4
	<i>U</i> <i>I</i> 30 <i>Q</i> 30 <i>C</i> 3	13	0.0	0.2	13	0	0.0	3.7	13	0	0.0	9.3	12	3	0.0	549.7	7	292	0.0	5.2
	<i>U</i> <i>I</i> 30 <i>Q</i> 30 <i>C</i> 4	15	0.0	0.2	15	0	0.0	3.5	15	0	0.0	12.1	13	4	0.0	278.8	7	211	0.0	7.7
	<i>U</i> <i>I</i> 30 <i>Q</i> 30 <i>C</i> 5	15	0.0	0.3	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	<i>U</i> <i>I</i> 50 <i>Q</i> 5 <i>C</i> 1	34	0.0	122.4	34	0	0.0	662.5	34	0	0.0	811.6	34	0	2.4	3601.2	34	71	0.0	170.6
	<i>U</i> <i>I</i> 50 <i>Q</i> 5 <i>C</i> 2	38	0.0	851.4	38	0	0.0	1671.9	38	0	0.0	1564.4	38	0	1.6	3600.2	38	108	0.0	57.8
	<i>U</i> <i>I</i> 50 <i>Q</i> 5 <i>C</i> 3	38	2.6	3600.1	38	0	2.6	3600.3	38	0	2.6	3600.0	38	0	2.6	3607.1	37	111	0.0	256.9
	<i>U</i> <i>I</i> 50 <i>Q</i> 5 <i>C</i> 4	37	2.7	3600.0	37	0	2.7	3600.1	37	0	2.7	3601.0	37	0	2.7	3600.2	36	89	0.0	103.0
	<i>U</i> <i>I</i> 50 <i>Q</i> 5 <i>C</i> 5	39	3.1	3602.1	39	0	3.1	3603.7	38	0	0.0	2525.5	38	1	1.2	3600.2	38	67	0.0	126.5
	<i>U</i> <i>I</i> 50 <i>Q</i> 10 <i>C</i> 1	18	0.0	159.1	18	0	0.0	608.7	18	0	0.0	978.5	18	0	0.0	2653.1	17	134	0.0	999.8
	<i>U</i> <i>I</i> 50 <i>Q</i> 10 <i>C</i> 2	20	0.0	569.7	20	0	0.0	1448.7	20	0	0.0	3561.3	20	0	6.5	3600.3	19	106	0.0	890.8
	<i>U</i> <i>I</i> 50 <i>Q</i> 10 <i>C</i> 3	20	0.0	226.2	20	0	0.0	641.5	20	0	0.0	1522.0	20	0	7.5	3600.5	19	118	0.0	271.4
	<i>U</i> <i>I</i> 50 <i>Q</i> 10 <i>C</i> 4	20	0.0	328.1	20	0	0.0	1393.8	20	0	0.0	2345.5	19	2	6.9	3600.4	18	75	0.0	736.6
	<i>U</i> <i>I</i> 50 <i>Q</i> 10 <i>C</i> 5	21	0.0	530.7	21	0	0.0	1327.3	21	0	0.0	2611.0	21	0	9.2	3600.7	19	175	0.0	399.0
	<i>U</i> <i>I</i> 50 <i>Q</i> 50 <i>C</i> 1	15	0.0	0.4	15	0	0.0	10.4	15	0	0.0	38.9	13	5	0.0	1051.1	7	361	0.0	20.1
	<i>U</i> <i>I</i> 50 <i>Q</i> 50 <i>C</i> 2	15	0.0	0.4	15	0	0.0	10.2	15	0	0.0	55.5	15	0	0.0	2467.9	7	415	0.0	23.7
	<i>U</i> <i>I</i> 50 <i>Q</i> 50 <i>C</i> 3	15	0.0	0.4	15	0	0.0	9.3	15	0	0.0	167.3	13	6	0.0	821.5	7	594	0.0	15.1
	<i>U</i> <i>I</i> 50 <i>Q</i> 50 <i>C</i> 4	15	0.0	0.4	15	0	0.0	11.3	15	0	0.0	69.7	14	3	0.0	1821.7	7	422	0.0	14.7
	<i>U</i> <i>I</i> 50 <i>Q</i> 50 <i>C</i> 5	15	0.0	0.5	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.2: Detailed CPLEX results for the MSSP formulations *PS* and *AS* (U-instances)<sup>1,2</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 2.

<sup>2</sup>Entries with less than 3600 seconds in the *Time* column indicate that the corresponding values are optimal.

### Appendix C. Detailed heuristic results for the MSSP

Instance	PS			AS																
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$							
	z	e	HGap Time	z	e	HGap Time	z	e	HGap Time	z	e	HGap Time	z	e	HGap Time					
I10	D_I10_Q5_C1	6	0.00	2.6	6	0	0.00	2.3	6	0	0.00	2.0	6	0	0.00	1.7	6	0	0.00	2.0
	D_I10_Q5_C2	12	0.00	1.4	12	0	0.00	3.3	12	0	0.00	3.6	10	2	0.00	2.6	8	17	0.00	2.3
	D_I10_Q5_C3	9	0.00	2.5	9	0	0.00	2.2	9	0	0.00	2.2	8	1	0.00	2.3	8	3	0.00	1.6
	D_I10_Q5_C4	12	0.00	1.4	12	0	0.00	4.0	12	0	0.00	4.1	11	1	0.00	3.6	10	47	0.00	2.4
	D_I10_Q5_C5	12	0.00	1.8	12	0	0.00	3.2	12	0	0.00	3.0	11	2	0.00	2.4	11	2	0.00	2.4
	D_I10_Q10_C1	5	0.00	3.5	5	0	0.00	1.9	5	0	0.00	1.8	5	0	0.00	2.0	4	5	0.00	2.1
	D_I10_Q10_C2	10	0.00	2.1	10	0	0.00	3.2	10	0	0.00	3.5	9	2	0.00	2.8	7	72	0.00	2.7
	D_I10_Q10_C3	5	0.00	3.6	5	0	0.00	2.0	5	0	0.00	2.0	4	3	0.00	1.5	4	3	0.00	1.5
	D_I10_Q10_C4	10	0.00	2.0	10	0	0.00	3.7	10	0	0.00	3.9	9	3	0.00	12.8	7	69	0.00	2.5
	D_I10_Q10_C5	10	0.00	3.6	10	0	0.00	3.7	10	0	0.00	4.0	10	0	0.00	5.1	7	51	0.00	2.4
I30	D_I30_Q5_C1	23	0.00	12.3	23	0	0.00	22.9	23	0	0.00	31.7	23	0	0.00	12.9	22	66	0.00	31.1
	D_I30_Q5_C2	29	0.00	11.0	29	0	0.00	54.8	29	0	0.00	180.8	28	3	0.88	518.0	27	79	0.00	26.4
	D_I30_Q5_C3	28	0.00	6.9	28	0	0.00	20.4	28	0	0.00	41.7	27	3	2.78	487.0	26	43	0.00	25.8
	D_I30_Q5_C4	30	0.00	8.4	30	0	-3.23	22.8	30	0	0.00	38.2	30	1	4.31	66.3	27	65	0.00	27.6
	D_I30_Q5_C5	30	0.00	7.2	30	0	0.00	22.4	30	0	0.00	39.8	29	1	0.00	23.4	28	17	0.00	26.5
	D_I30_Q10_C1	14	0.00	12.5	14	0	0.00	24.7	14	0	0.00	39.7	14	2	3.57	35.6	12	144	0.00	64.8
	D_I30_Q10_C2	17	0.00	24.7	17	0	0.00	18.3	17	0	0.00	51.2	17	0	3.03	88.3	14	51	0.00	19.8
	D_I30_Q10_C3	15	0.00	11.4	15	0	0.00	19.3	15	0	0.00	33.9	15	0	0.00	16.1	14	66	0.00	23.1
	D_I30_Q10_C4	18	0.00	17.2	18	0	0.00	22.5	18	0	0.00	38.7	17	0	0.00	51.1	15	56	0.00	20.5
	D_I30_Q10_C5	15	0.00	12.7	15	0	0.00	18.9	15	0	0.00	37.6	15	3	5.00	27.3	15	20	0.00	22.5
	D_I30_Q30_C1	10	0.00	16.6	10	0	0.00	27.6	10	0	0.00	29.7	10	0	0.00	29.8	7	207	0.00	56.8
	D_I30_Q30_C2	10	0.00	16.5	10	0	0.00	19.0	10	0	0.00	29.5	10	0	0.00	41.1	7	126	0.00	21.6
	D_I30_Q30_C3	10	0.00	9.8	10	0	0.00	21.0	10	0	0.00	29.3	10	0	0.00	30.7	7	234	0.00	21.6
	D_I30_Q30_C4	10	0.00	11.4	10	0	0.00	26.2	10	0	0.00	29.7	10	0	0.00	43.4	7	177	0.00	21.2
	D_I30_Q30_C5	10	0.00	14.3	10	0	0.00	27.7	10	0	0.00	27.2	10	0	0.00	37.4	7	255	0.00	20.9
I50	D_I50_Q5_C1	44	-2.22	17.6	44	0	-2.22	12.2	44	0	-2.22	47.3	44	0	0.57	40.3	43	95	0.00	41.0
	D_I50_Q5_C2	47	0.00	17.9	47	0	0.00	13.8	47	0	0.00	18.4	46	2	-0.53	41.7	45	135	0.00	50.8
	D_I50_Q5_C3	49	0.00	52.8	49	0	0.00	31.5	49	0	0.00	57.5	48	2	-0.51	69.6	47	117	0.00	50.8
	D_I50_Q5_C4	50	0.00	24.0	50	0	0.00	56.6	50	0	0.00	95.9	49	0	0.00	77.9	48	104	0.00	39.1
	D_I50_Q5_C5	52	0.00	39.1	52	0	0.00	27.3	52	0	0.00	46.9	51	1	0.00	61.3	51	2	0.00	43.9
	D_I50_Q10_C1	25	0.00	57.1	25	0	0.00	47.1	25	0	0.00	65.6	24	3	2.06	20.3	22	138	0.00	37.1
	D_I50_Q10_C2	24	0.00	72.5	24	0	0.00	64.7	24	0	0.00	64.3	24	0	0.00	68.2	23	123	0.00	43.6
	D_I50_Q10_C3	27	0.00	58.4	27	0	0.00	58.5	27	0	0.00	54.9	26	3	3.88	139.2	24	100	0.00	62.9
	D_I50_Q10_C4	28	0.00	41.8	28	0	0.00	66.3	27	0	0.00	47.7	25	2	0.00	52.2	24	17	0.00	44.5
	D_I50_Q10_C5	28	0.00	52.4	28	0	0.00	48.7	28	0	0.00	50.4	29	1	4.46	165.5	26	107	0.00	34.2
	U_I50_Q50_C1	10	0.00	60.4	10	0	0.00	29.1	10	0	0.00	68.4	10	0	0.00	58.0	7	360	0.00	72.4
	U_I50_Q50_C2	10	0.00	76.6	10	0	0.00	20.8	10	0	0.00	83.0	10	0	0.00	81.4	7	273	0.00	70.1
	U_I50_Q50_C3	10	0.00	59.5	10	0	0.00	29.8	10	0	0.00	64.1	10	0	0.00	67.7	7	201	0.00	63.7
	U_I50_Q50_C4	10	0.00	43.5	10	0	0.00	19.9	10	0	0.00	68.3	10	0	0.00	75.8	7	279	0.00	52.7
	U_I50_Q50_C5	10	0.00	53.8	10	0	0.00	19.0	10	0	0.00	80.9	10	0	0.00	46.7	7	243	0.00	54.5

Table C.1: Heuristic results for the MSSP (D-instances) <sup>1</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 4.

Instance	PS			AS																
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$							
	<i>z</i>	<i>HGap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>	<i>z</i>	<i>e</i>	<i>HGap</i>	<i>Time</i>					
I10	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 5_ <i>C</i> 1	7	0.00	1.6	7	0	0.00	3.0	7	0	0.00	3.4	6	2	0.00	1.9	5	15	0.00	1.7
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 5_ <i>C</i> 2	13	0.00	1.6	13	0	0.00	3.6	13	0	0.00	3.6	11	1	0.00	2.6	8	52	0.00	2.2
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 5_ <i>C</i> 3	12	0.00	2.5	12	0	0.00	3.3	12	0	0.00	3.5	11	2	0.00	4.2	9	28	0.00	2.2
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 5_ <i>C</i> 4	13	0.00	1.6	13	0	0.00	3.4	13	0	0.00	3.9	11	3	0.00	3.1	9	25	0.00	2.2
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 5_ <i>C</i> 5	10	0.00	1.9	10	0	0.00	2.8	10	0	0.00	3.2	10	0	0.00	3.0	8	19	0.00	2.1
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 10_ <i>C</i> 1	7	0.00	4.6	7	0	0.00	2.8	7	0	0.00	2.8	5	2	0.00	2.5	3	27	0.00	1.7
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 10_ <i>C</i> 2	13	0.00	2.5	13	0	0.00	4.1	13	0	0.00	4.2	11	1	0.00	3.3	7	51	0.00	2.8
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 10_ <i>C</i> 3	12	0.00	4.1	12	0	0.00	3.9	12	0	0.00	4.2	11	0	0.00	4.0	7	34	0.00	2.7
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 10_ <i>C</i> 4	13	0.00	2.4	13	0	0.00	4.0	13	0	0.00	4.5	11	3	0.00	3.4	7	35	0.00	2.7
	<i>U</i> _ <i>I</i> 10_ <i>Q</i> 10_ <i>C</i> 5	10	0.00	5.3	10	0	0.00	3.1	10	0	0.00	3.7	9	2	0.00	15.9	6	22	0.00	2.4
I30	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 5_ <i>C</i> 1	18	0.00	7.4	18	0	0.00	16.8	18	0	0.00	47.6	18	2	2.78	33.7	17	59	0.00	25.4
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 5_ <i>C</i> 2	22	0.00	8.1	22	0	0.00	94.1	22	0	0.00	107.8	22	1	4.71	129.0	21	83	0.00	28.9
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 5_ <i>C</i> 3	22	0.00	9.6	22	0	0.00	21.1	22	0	0.00	44.3	22	2	2.27	54.0	22	76	0.00	56.5
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 5_ <i>C</i> 4	24	0.00	17.4	24	0	0.00	22.3	24	0	0.00	44.6	24	0	0.00	29.9	24	23	0.00	33.5
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 5_ <i>C</i> 5	23	0.00	5.1	23	0	0.00	84.0	23	0	0.00	47.0	23	3	5.56	46.9	22	81	0.00	36.3
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 10_ <i>C</i> 1	12	0.00	16.8	12	0	0.00	24.8	12	0	0.00	57.2	11	2	0.00	55.9	9	69	0.00	51.6
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 10_ <i>C</i> 2	15	0.00	49.0	15	0	0.00	15.8	15	0	0.00	61.8	14	0	0.00	32.7	11	99	0.00	20.7
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 10_ <i>C</i> 3	14	0.00	22.4	14	0	0.00	21.2	14	0	0.00	37.0	13	3	1.85	29.2	11	65	0.00	23.1
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 10_ <i>C</i> 4	15	0.00	75.4	15	0	0.00	19.5	15	0	0.00	124.7	14	3	5.36	22.5	12	67	0.00	27.5
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 10_ <i>C</i> 5	15	0.00	48.8	15	0	0.00	19.8	15	0	0.00	51.9	15	0	5.26	29.4	12	84	0.00	26.1
I50	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 30_ <i>C</i> 1	12	0.00	17.5	12	0	0.00	36.1	12	0	0.00	96.6	11	0	0.00	76.4	6	173	0.00	28.6
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 30_ <i>C</i> 2	15	0.00	22.7	15	0	0.00	17.0	15	0	0.00	43.7	14	0	0.00	31.0	7	157	0.00	37.0
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 30_ <i>C</i> 3	13	0.00	24.1	13	0	0.00	31.0	13	0	0.00	54.3	12	3	0.00	45.5	7	167	0.00	65.4
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 30_ <i>C</i> 4	15	0.00	17.1	15	0	0.00	17.0	15	0	0.00	48.9	14	3	5.36	27.8	7	145	0.00	54.8
	<i>U</i> _ <i>I</i> 30_ <i>Q</i> 30_ <i>C</i> 5	15	0.00	52.8	15	0	0.00	16.2	15	0	0.00	40.2	14	3	5.36	33.2	7	129	0.00	59.1
	<i>U</i> _ <i>I</i> 50_ <i>Q</i> 5_ <i>C</i> 1	34	0.00	51.8	34	0	0.00	201.4	34	0	0.00	70.0	34	0	0.00	10.7	34	103	0.00	74.9
	<i>U</i> _ <i>I</i> 50_ <i>Q</i> 5_ <i>C</i> 2	38	0.00	75.8	38	0	0.00	70.0	38	0	0.00	71.6	39	1	3.29	8.4	38	41	0.00	44.0
	<i>U</i> _ <i>I</i> 50_ <i>Q</i> 5_ <i>C</i> 3	38	0.00	52.8	38	0	0.00	84.4	38	0	0.00	121.0	37	0	-2.63	8.5	37	156	0.00	51.1
	<i>U</i> _ <i>I</i> 50_ <i>Q</i> 5_ <i>C</i> 4	37	0.00	24.0	37	0	0.00	104.1	37	0	0.00	89.6	36	0	-2.70	6.9	36	78	0.00	40.5
	<i>U</i> _ <i>I</i> 50_ <i>Q</i> 5_ <i>C</i> 5	38	-2.56	39.1	38	0	-2.56	83.1	38	0	0.00	61.7	38	0	-0.65	12.7	38	151	0.00	47.4
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 10_ <i>C</i> 1	18	0.00	57.6	18	0	0.00	45.9	18	0	0.00	54.0	18	0	0.00	168.7	17	63	0.00	72.1	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 10_ <i>C</i> 2	20	0.00	95.3	20	0	0.00	86.5	20	0	0.00	58.4	21	0	5.00	184.2	19	123	0.00	57.3	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 10_ <i>C</i> 3	20	0.00	70.4	20	0	0.00	86.7	20	0	0.00	108.7	20	3	3.75	135.1	19	72	0.00	67.5	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 10_ <i>C</i> 4	20	0.00	58.5	20	0	0.00	47.0	20	0	0.00	55.3	20	0	2.56	154.4	18	139	0.00	42.4	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 10_ <i>C</i> 5	21	0.00	96.4	21	0	0.00	75.6	21	0	0.00	51.9	20	2	-2.38	118.4	19	93	0.00	71.4	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 50_ <i>C</i> 1	15	0.00	61.9	15	0	0.00	39.7	15	0	0.00	162.8	14	2	1.75	138.1	7	159	0.00	101.3	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 50_ <i>C</i> 2	15	0.00	64.3	15	0	0.00	41.4	15	0	0.00	100.8	15	3	5.00	155.8	7	252	0.00	71.1	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 50_ <i>C</i> 3	15	0.00	47.4	15	0	0.00	42.7	15	0	0.00	230.4	14	2	0.00	293.5	7	234	0.00	136.5	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 50_ <i>C</i> 4	15	0.00	40.1	15	0	0.00	42.2	15	0	0.00	349.8	14	3	0.00	190.0	7	212	0.00	196.0	
<i>U</i> _ <i>I</i> 50_ <i>Q</i> 50_ <i>C</i> 5	15	0.00	67.6	15	0	0.00	42.1	15	0	0.00	253.1	15	2	5.08	318.5	7	197	0.00	147.6	

Table C.2: Heuristic results for the MSSP (U-instances) <sup>1</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 4.

Instance	PS			AS												
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$			
	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	
I10	D_I10_Q5_C1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	D_I10_Q5_C2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38.10	4.76	0.88	25.00
	D_I10_Q5_C3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.06	0.00	0.00	0.00
	D_I10_Q5_C4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	0.00	0.00	0.00
	D_I10_Q5_C5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	D_I10_Q10_C1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00
	D_I10_Q10_C2	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	15.79	0.00	0.00	0.00	0.00	0.00
	D_I10_Q10_C3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	D_I10_Q10_C4	10.00	0.00	0.00	10.00	0.00	0.00	10.00	0.00	0.00	10.26	0.00	0.00	0.00	0.00	0.00
	D_I10_Q10_C5	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	2.50	0.00	0.00	0.00	0.00	0.00
I30	D_I30_Q5_C1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.70	0.00	0.00	4.55	4.55	0.00	
	D_I30_Q5_C2	6.90	0.00	0.00	3.45	0.00	0.00	6.90	0.00	0.00	21.93	0.88	0.88	3.70	3.70	
	D_I30_Q5_C3	3.57	0.00	0.00	3.57	0.00	0.00	3.57	0.00	0.00	14.81	2.78	2.78	3.85	3.85	
	D_I30_Q5_C4	6.67	3.33	0.00	3.23	0.00	-3.23	6.67	0.00	0.00	23.28	4.31	4.31	11.11	7.41	
	D_I30_Q5_C5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.56	0.00	0.00	0.00	0.00	0.00	
	D_I30_Q10_C1	14.29	7.14	0.00	7.14	0.00	0.00	14.29	0.00	0.00	50.00	3.57	3.57	8.33	0.00	
	D_I30_Q10_C2	11.76	5.88	0.00	5.88	0.00	0.00	11.76	5.88	0.00	51.52	3.03	3.03	7.14	7.14	
	D_I30_Q10_C3	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	16.67	0.00	0.00	0.00	0.00	
	D_I30_Q10_C4	5.56	0.00	0.00	5.56	0.00	0.00	5.56	0.00	0.00	60.29	0.00	0.00	13.33	0.00	
	D_I30_Q10_C5	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	10.00	5.00	5.00	0.00	0.00	
I50	D_I50_Q5_C1	10.00	0.00	0.00	10.00	0.00	0.00	10.00	0.00	0.00	52.50	0.00	0.00	0.00	0.00	
	D_I50_Q5_C2	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	65.00	0.00	0.00	0.00	0.00	
	D_I50_Q5_C3	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	45.00	0.00	0.00	0.00	0.00	
	D_I50_Q5_C4	10.00	0.00	0.00	10.00	0.00	0.00	10.00	0.00	0.00	37.50	25.00	0.00	0.00	0.00	
	D_I50_Q5_C5	10.00	0.00	0.00	10.00	0.00	0.00	10.00	0.00	0.00	42.50	35.00	0.00	0.00	0.00	
	D_I50_Q10_C1	8.00	4.00	0.00	4.00	0.00	0.00	8.00	0.00	0.00	56.70	2.06	2.06	4.55	4.55	
	D_I50_Q10_C2	4.17	0.00	0.00	4.17	0.00	0.00	4.17	0.00	0.00	19.79	0.00	0.00	4.35	4.35	
	D_I50_Q10_C3	3.70	0.00	0.00	3.70	0.00	0.00	3.70	0.00	0.00	46.60	3.88	3.88	8.33	8.33	
	D_I50_Q10_C4	7.14	3.57	0.00	3.57	0.00	0.00	11.11	7.41	0.00	53.92	0.00	0.00	8.33	8.33	
	D_I50_Q10_C5	7.14	3.57	0.00	3.57	0.00	0.00	7.14	3.57	0.00	19.64	4.46	4.46	3.85	3.85	
U	U_I50_Q50_C1	10.00	0.00	0.00	10.00	0.00	0.00	10.00	0.00	0.00	117.50	10.00	0.00	0.00	0.00	
	U_I50_Q50_C2	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	67.50	37.50	0.00	0.00	0.00	
	U_I50_Q50_C3	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	57.50	25.00	0.00	0.00	0.00	
	U_I50_Q50_C4	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	62.50	40.00	0.00	0.00	0.00	
	U_I50_Q50_C5	10.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00	52.50	35.00	0.00	0.00	0.00	

Table C.3: Relative contribution of each heuristic phase (D-instances)<sup>1</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 5.

Instance	PS			AS												
				$\beta = 0.2$			$\beta = 0.5$			$\beta = 0.8$			$\beta = 1.0$			
	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	HG <sub>G</sub>	HG <sub>LS</sub>	HG <sub>P</sub>	
I10	U_I10_Q5_C1	14.29	0.00	0.00	14.29	0.00	0.00	14.29	0.00	0.00	34.62	0.00	0.00	0.00	0.00	0.00
	U_I10_Q5_C2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.22	0.00	0.00	12.50	12.50	0.00
	U_I10_Q5_C3	8.33	0.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00	23.91	0.00	0.00	0.00	0.00	0.00
	U_I10_Q5_C4	15.38	0.00	0.00	15.38	0.00	0.00	15.38	0.00	0.00	2.13	0.00	0.00	0.00	0.00	0.00
	U_I10_Q5_C5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	27.50	0.00	0.00	0.00	0.00	0.00
	U_I10_Q10_C1	14.29	0.00	0.00	14.29	0.00	0.00	14.29	0.00	0.00	40.91	0.00	0.00	0.00	0.00	0.00
	U_I10_Q10_C2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	U_I10_Q10_C3	8.33	0.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00	6.82	0.00	0.00	0.00	0.00	0.00
	U_I10_Q10_C4	15.38	0.00	0.00	15.38	0.00	0.00	15.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	U_I10_Q10_C5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13.16	0.00	0.00	0.00	0.00	0.00
I30	U_I30_Q5_C1	11.11	5.56	0.00	5.56	5.56	0.00	11.11	5.56	0.00	27.78	2.78	2.78	5.88	5.88	0.00
	U_I30_Q5_C2	9.09	0.00	0.00	9.09	0.00	0.00	9.09	0.00	0.00	68.24	4.71	4.71	4.76	4.76	0.00
	U_I30_Q5_C3	9.09	4.55	0.00	9.09	4.55	0.00	9.09	9.09	0.00	23.86	2.27	2.27	4.55	0.00	0.00
	U_I30_Q5_C4	4.17	0.00	0.00	8.33	0.00	0.00	8.33	4.17	0.00	23.96	0.00	0.00	0.00	0.00	0.00
	U_I30_Q5_C5	4.35	4.35	0.00	13.04	4.35	0.00	4.35	4.35	0.00	50.00	5.56	5.56	4.55	4.55	0.00
	U_I30_Q10_C1	16.67	8.33	0.00	8.33	8.33	0.00	16.67	8.33	0.00	41.30	15.22	0.00	22.22	0.00	0.00
	U_I30_Q10_C2	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	75.00	7.14	0.00	9.09	9.09	0.00
	U_I30_Q10_C3	7.14	0.00	0.00	7.14	0.00	0.00	7.14	7.14	0.00	38.89	1.85	1.85	18.18	9.09	0.00
	U_I30_Q10_C4	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	76.79	8.93	5.36	16.67	8.33	0.00
	U_I30_Q10_C5	13.33	0.00	0.00	13.33	0.00	0.00	13.33	6.67	0.00	82.46	5.26	5.26	8.33	0.00	0.00
I50	U_I50_Q5_C1	5.88	0.00	0.00	11.76	0.00	0.00	5.88	5.88	0.00	36.76	0.00	0.00	2.94	2.94	0.00
	U_I50_Q5_C2	7.89	2.63	0.00	7.89	2.63	0.00	7.89	5.26	0.00	26.97	3.29	3.29	0.00	0.00	0.00
	U_I50_Q5_C3	7.89	2.63	0.00	5.26	2.63	0.00	10.53	5.26	0.00	30.92	0.00	-2.63	2.70	2.70	0.00
	U_I50_Q5_C4	2.70	2.70	0.00	8.11	2.70	0.00	8.11	2.70	0.00	26.35	0.00	-2.70	5.56	5.56	0.00
	U_I50_Q5_C5	5.13	0.00	-2.56	5.13	0.00	-2.56	7.89	7.89	0.00	40.52	2.61	-0.65	2.63	2.63	0.00
	U_I50_Q10_C1	16.67	5.56	0.00	11.11	11.11	0.00	16.67	11.11	0.00	81.94	5.56	0.00	11.76	5.88	0.00
	U_I50_Q10_C2	10.00	5.00	0.00	15.00	10.00	0.00	10.00	10.00	0.00	153.75	5.00	5.00	10.53	5.26	0.00
	U_I50_Q10_C3	15.00	5.00	0.00	15.00	5.00	0.00	15.00	5.00	0.00	85.00	3.75	3.75	0.00	0.00	0.00
	U_I50_Q10_C4	15.00	10.00	0.00	10.00	0.00	0.00	15.00	15.00	0.00	141.03	2.56	2.56	16.67	11.11	0.00
	U_I50_Q10_C5	9.52	4.76	0.00	4.76	0.00	0.00	9.52	9.52	0.00	80.95	2.38	-2.38	5.26	5.26	0.00
I50	U_I50_Q50_C1	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	50.88	1.75	1.75	0.00	0.00	0.00
	U_I50_Q50_C2	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	120.00	11.67	5.00	0.00	0.00	0.00
	U_I50_Q50_C3	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	60.34	8.62	0.00	0.00	0.00	0.00
	U_I50_Q50_C4	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	96.61	20.34	0.00	0.00	0.00	0.00
	U_I50_Q50_C5	6.67	0.00	0.00	6.67	0.00	0.00	6.67	0.00	0.00	88.14	13.56	5.08	0.00	0.00	0.00

Table C.4: Relative contribution of each heuristic phase (U-instances)<sup>1</sup>.

<sup>1</sup>The meaning of the headings of the columns is as in Table 5.