European Journal of Operational Research 251 (2016) 751–770

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Scheduling policies for multi-period services

Cristina Núñez-del-Toro[a,*], Elena Fernández[a], Jörg Kalcsics[b], Stefan Nickel[c]

[a] Statistics and Operations Research Department, Universitat Politècnica de Catalunya, Barcelona, Spain
[b] School of Mathematics, University of Edinburgh, Edinburgh, Scotland
[c] Institute of Operations Research, Karlsruhe Institute of Technology, Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 12 February 2015
Accepted 2 December 2015
Available online 30 December 2015

Keywords:
Combinatorial optimization
Multi-period problems
Service scheduling
Heuristics

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.

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 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 Fernández, Kalcsics, Nickel, and Ríos-Mercado (2010); Queiruga, Walther, González-Benito, and Spengler (2008). 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 (2007).

The rest of the paper is organized as follows. In the next section we formally describe our problem and its modeling 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

[*] Corresponding author. Tel.: +34 934054095.
E-mail address: cristina.nunez@upc.edu (C. Núñez-del-Toro).

http://dx.doi.org/10.1016/j.ejor.2015.12.002
0377-2217/© 2015 Elsevier B.V. All rights reserved.

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 $C_i = \{t_i^1, \ldots, t_i^{|C_i|}\} \subseteq T$ 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 $C_i = \{t_i^1, \ldots, t_i^{|C_i|}\} \subseteq T$ 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.

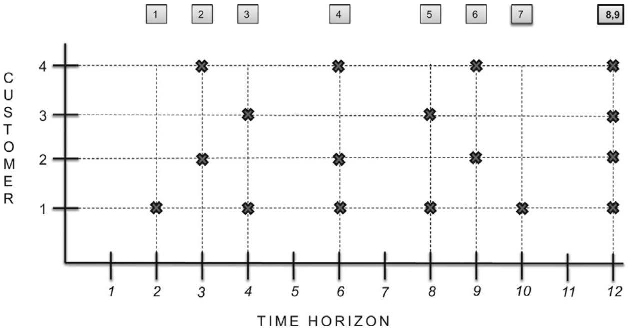

Fig. 1. Solution of the example for the periodic service policy.

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. Fig. 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.

However, under the AS policy we just need eight operators instead of nine, see Fig. 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.

A main aim of this paper is to compare the two different service policies. We are especially interested in the trade off between