INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 0 (2022) 1–31
DOI: 10.1111/itor.13127

Hybridizing adaptive large neighborhood search with kernel search: a new solution approach for the nurse routing problem with incompatible services and minimum demand

Alessandro Gobbi[a,*], Daniele Manerba[b], Renata Mansini[b] and Roberto Zanotti[b]

[a] *Department of Clinical and Experimental Sciences, University of Brescia, viale Europa 11, Brescia 25123, Italy*
[b] *Department of Information Engineering, University of Brescia, via Branze 38, Brescia 25123, Italy*
E-mail: alessandro.gobbi@unibs.it [Gobbi]; daniele.manerba@unibs.it [Manerba]; renata.mansini@unibs.it [Mansini]; roberto.zanotti@unibs.it [Zanotti]

Received 28 December 2020; received in revised form 18 October 2021; accepted 26 January 2022

Abstract

The average age of the population has grown steadily in recent decades along with the number of people suffering from chronic diseases and asking for treatments. Hospital care is expensive and often unsafe, especially for older individuals. This is particularly true during pandemics as the recent SARS-CoV-2. Hospitalization at home has become a valuable alternative to face efficiently a huge increase in treatment requests while guaranteeing a high quality of service and lower risk to fragile patients. This new model of care requires the redefinition of health services organization and the optimization of scarce resources (e.g., available nurses). In this paper, we study a Nurse Routing Problem that tries to find a good balance between hospital costs reduction and the well-being of patients, also considering realistic operational restrictions like maximum working times for the nurses and possible incompatibilities between services jointly provided to the same patient. We first propose a Mixed Integer Linear Programming formulation for the problem and use some valid inequalities to strengthen it. A simple branch-and-cut algorithm is proposed and validated to derive ground benchmarks. In addition, to efficiently solve the problem, we develop an Adaptive Large Neighborhood Search hybridized with a Kernel Search and validate its performance over a large set of different realistic working scenarios. Computational tests show how our matheuristic approach manages to find good solutions in a reasonable amount of time even in the most difficult settings. Finally, some interesting managerial insights are discussed through an economic analysis of the operating context.

Keywords: home healthcare; nurse routing problem; incompatible services; ALNS; kernel search

*Corresponding author.

1. Introduction

The constant average aging of the world population (Department of Economic and Social Affairs of the United Nations Secretariat, 2017) has led private and public organizations to face new challenges in the field of personal healthcare, and this trend is going to continue in the future. In particular, a sizable proportion of people is more prone to chronic diseases typical of old age, such as diabetes or dementia, or to physical impairments. These diseases very often force the relatives of patients to hospitalize them in specialized centers despite the fact this solution is not well received by both sides (Nolan, 1999). For this reason, more and more people are interested in applying for home healthcare (HHC) services (see, e.g., Landers et al., 2016). Actually, HHC covers an even wider set of needs, not necessarily associated with old age, and making hospitalization at home a valuable alternative. Required services range from simple assistance and support (dressing and bathing), to specific medical operations such as administering drugs, performing injections, and measuring vital signs (e.g., temperature and blood pressure) or even assisting the patient throughout his/her chemotherapy plan (Chahed et al., 2009). Although this has been a challenging and profitable sector for private healthcare companies, recently also public organizations have realized that providing these services with appropriate frequency and quality can reduce costs and increase patient satisfaction. Hospitalization at home may even become a necessity to protect frail patients from contamination during pandemic events.

In this work, we study a Nurse Routing Problem (NRP) where a set of nurses have to be routed and scheduled to visit a set of patients spread over a geographical area to perform care services. Each patient may require more than one service type. A profit (not necessarily associated with an economical reward but possibly representing the relevance or urgency) and a service time are associated with each service request. Moreover, each nurse has a predefined working time that cannot be exceeded considering both the time required to serve the patients and the traveling time. The problem aims at selecting the subset of requests to fulfill that maximizes the total profit collected while complying with nurse time constraints. To make our operational setting more realistic, two additional constraints are introduced: (i) *services incompatibility*, imposing that if any two service requests are incompatible they cannot be performed to the same patient on the same day, and (ii) *minimum demand satisfaction*, requiring that a minimum number of requests for each service type has to be satisfied. In this paper, we analyze two variants of NRP: the first one, called NRP with Incompatible Services (NRP-IS), takes into account only the incompatibility constraints, whereas the second one, called NRP-IS with Demand (NRP-ISD), adds minimum demand satisfaction to the previous one.

In the scientific literature, incompatibility constraints have been largely studied under different names as *negative disjunctive*, *conflicts* or *exclusionary side constraints*. Examples of their applications arise in the context of horizontal collaboration among carriers and shippers where transportation lanes cannot include incompatible goods as food and chemicals (see Colombi et al., 2017), in transportation problems where pairs of suppliers are incompatible when serving the same customer (Goossens and Spieksma, 2009), knapsack problems (Bettinelli et al., 2017), and max-flow problems (Şuvak et al., 2020). The introduction of services incompatibility in the healthcare domain has a high practical relevance. Nowadays, existing healthcare systems try to concentrate services in order to disturb patients as little as possible. Potentially, this implies that if a nurse has enough time and if different services are required by the same patient, these can be executed in sequence in only one visit. Unfortunately, although possible in practice, there are specific services that should not or even must not, be jointly executed (*incompatible services*). The reasons for considering services incompatibility can derive from a specific care giver policy, may be induced by the fact that a frail patient cannot endure some intensive care services on the same day, or may even be imposed from potentially harmful interactions between services. Several real case examples can be mentioned about such incompatibilities. For instance, an elderly person would not be able to receive a transfusion and a hygiene service one after the other, or a critically ill patient receiving local cleansing of a skin ulcer cannot be treated with an anticoagulant drug in the same day. Moreover, there are some drugs that cannot be infused simultaneously and not even in sequence, but a minimum time interval of at least one day has to elapse between them. However, despite their practical relevance, incompatibilities do not explicitly appear in many works in the healthcare domain (see Section 2).

Finally, the presence of profits associated with the requests and the services incompatibility make the studied problems significantly different from the ones dealt with in the healthcare scientific area which are commonly modeled as Vehicle Routing Problems (VRPs). Classical VRPs are operational models that need a precise selection of the patient requests to satisfy in advance and focus on minimizing the overall service cost (or time, or any proportional measure). Instead, we model the NRP as a team orienteering problem (Vansteenwegen and Gunawan, 2019), in which the selection of the requests to satisfy is a critical decision. Moreover, this allows us to consider the optimization of measures directly related to the quality of service, represented as profits to maximize. Furthermore, since we are considering different service types, our NRP can be seen as a variant of the MVTPP (Manerba and Mansini, 2015; Gendreau et al., 2016; Manerba et al., 2017), as originally noted in Manerba and Mansini (2016).

The paper provides some relevant contributions. First, a compact mathematical model to formalize the NRP-IS is defined and some simple valid inequalities to strengthen it are proposed. This allows us to devise a simple branch-and-cut and a heuristic framework that takes advantage of the increased computing power of Mixed Integer Linear Programming (MILP) solvers. In fact, in order to tackle the problem efficiently also under realistic large-size instances, we develop a hybrid Adaptive Large Neighborhood Search (ALNS) embedding an intensification phase carried out by Kernel Search (KS) all the times that ALNS stalls without finding improving solutions. KS is a general-purpose framework that, in the spirit of exact neighborhoods exploration, exploits the compact formulation of the problem to build restricted problems and solves them by means of an MILP solver used as a black-box (see Angelelli et al., 2010, 2012). Third, experimental evidence of the viability of the proposed hybrid approach is provided. In particular, an extensive computational campaign, based on realistic randomly-generated instances, has shown how our method is efficient and effective, representing a valuable tool for supporting healthcare organizations in a day-by-day planning.

The paper is organized as follows. In Section 2, we summarize the recent literature related to routing and scheduling problems in HHC. In Section 3, we describe the NRP-IS in detail, propose a new MILP formulation along with some valid inequalities, and discuss its variant NRP-ISD. In Section 4, we propose the basic features of the developed ALNS approach, while in Section 5 the Kernel Search-based intensification procedure is described. The results of different computational tests on a large set of instances are presented in Section 6, while some managerial insights for the studied operational setting are derived in Section 7. Finally, Section 8 concludes the paper and sketches some future developments.

2. Literature review

HHC problems usually deal with planning the operations of a set of workers (typically, nurses) over a certain time horizon, that is, they assign each nurse a specific set of service requests (scheduling) to perform at the patients' home and establish the order in which to visit them (routing). For more details, the reader is referred to Fikar and Hirsch (2017) and Di Mascolo et al. (2017). In this paper, we focus on the most recently published works related to our problem and concerning deterministic single-period nurse routing and scheduling problems. Table 1 provides a classification of the surveyed articles according to the objective function used, the constraints introduced, and the solution method proposed.

It can be noticed that most of the problems aim at minimizing the overall traveling distances (TD), the traveling costs (TC), or times (TT), being considered as a natural extension of VRPs. Other more peculiar goals are the minimization of overtime (OT) and waiting times (WT). Only a few papers aim at minimizing the number of employed nurses (#N), probably because optimizing staff employment is a long-term goal, while in single-period planning staff is fixed. In addition to those related to costs, other factors are important in HHC operations such as service coverage, quality, and fairness. In many real working scenarios, the provider is required to visit each patient, but when this is not the case, maximizing the number of served requests (#R) could be a relevant objective. The overall patients satisfaction (PS), calculated as the percentage of fulfilled requests with respect to all those required by a patient, is frequently an important aspect to maximize (Gobbi et al., 2019). When dealing with multiple goals to optimize, some works employ weighted objective functions, including patient or nurse preferences (PR), various constraint violation (CV) penalties (e.g., related to time windows and skill assignments), or fairness oriented issues (FA) such as balanced workloads. Finally, there are some works that focus on very specific goals. Koeleman et al. (2012) minimize a combination of rejection and holding costs for patients, whereas Nasir and Dang (2018) define an objective function with eight components, including hiring costs of nurses and penalties if a patient has not been visited during the working day.

Concerning the modeling constraints, we can observe that time windows (TW), skill requirements (SK), and working time regulations (TR) are the most frequent. However, the specific implementations of such constraints vary substantially among the works. Regarding TW, we have to distinguish the use of hard and soft time windows. In the former case, services have to be provided without exceptions within the time window (the patients promptly require a life-saving medicine or injection, e.g., diabetic patients). In the latter, a violation of the time window is allowed by paying a penalty (the time preferences imposed by patient needs should be met). Not surprisingly, soft time windows are by far the most studied type of constraint. With regard to skill preferences, typically, a skill level is attributed to each nurse who will then be enabled or not to perform a particular service. Although in Trautsamwieser et al. (2011), Trautsamwieser and Hirsch (2011), and Hiermann et al. (2015) a nurse can perform all the services requiring lower skill levels, in Fikar and Hirsch (2015) a limit on such services is defined to prevent too many highly specialized services from remaining uncovered. TR usually concerns a limit on the working hours of a nurse or on the total distance

Table 1
Classification of NRPs

Paper	Objective function	Constraints	Solution approach(es)
Hindle et al. (2000)	TT / TD	TR	Approximation / Local search
Eveborn et al. (2006)	TC	TW / SK / BK / SY	Local search
Bertels and Fahle (2006)	TC / PR / CV	TW / SK / TR / BK	Matheuristic
Akjiratikarl et al. (2007)	TD	TW / TR	Metaheuristic
Bredstrom and Ronnqvist (2008)	TT / PR / FA	TW / SK / TR / HP / SY	Exact / Metaheuristic
Bräysy et al. (2009)	TD	TW / TR	Black-box commercial VRP solver
Dohn et al. (2009)	#R	TW / SK / SY	Exact
Hindle et al. (2009)	TC	TR	Approximation
Trautsamwieser and Hirsch (2011)	TT / WT / OT / PR / CV	TW / SK / TR	Exact / Metaheuristic
Trautsamwieser et al. (2011)	TT / WT / OT / PR / CV	TW / SK / TR	Exact / Metaheuristic
Bachouch et al. (2011)	TD	TW / SK	Exact
Rasmussen et al. (2012)	TC / PR / FA	TW/ SK / HP / SY	Exact
Rest et al. (2012)	TT / WT / OT	TW / SK / TR	Metaheuristic
Allaoua et al. (2013)	#N	TW / SK	Matheuristic
Mutingi and Mbohwa (2014)	TD / CV / FA	TW / SK / SY	Metaheurstic
Mankowska et al. (2014)	TC / FA / #R	TW / SK / HP / SY	Exact / Metaheuristic
Fikar and Hirsch (2015)	TT / WT	TW / SK / TR / BK / SY	Matheuristic
Hiermann et al. (2015)	TT / OT / PR / CV	TW / SK	Metaheuristic
Manerba and Mansini (2016)	PR	TR / IS	Exact
Redjem and Marcon (2016)	TT / WT	TW / HP	Local search
Yalçindağ et al. (2016)	TC / #N / CV	TR	Exact / Metaheuristic
Braekers et al. (2016)	TC / OT / PR / CV	TW / SK / TR	Exact / Metaheuristic
Nasir and Dang (2018)	TT	TW / SK / TR	Metaheuristic
Fathollahi-Fard et al. (2018)	TT	TW	Local search
Martinez et al. (2018)	TT / WT	TW / SK / TR	Local search
Di Mascolo et al. (2018)	CV	TW / SK / TR / SY	Exact
Gobbi et al. (2019)	PS	TW / TR	Matheuristic
Lasfargeas et al. (2019)	TT / WT	TW / SK / SY	Metaheuristic

traveled. In most of the works, these limitations are treated as daily time windows (e.g., eight hours), whereas Rest et al. (2012) study a scenario in which each nurse has more specific shifts to be respected on the same working day. Constraints that impose mandatory breaks during the day are less common. Bertels and Fahle (2006), Eveborn et al. (2006), and Bachouch et al. (2011) impose a mandatory visit to a dummy request representing a break, whereas other works (Trautsamwieser and Hirsch, 2011; Trautsamwieser et al., 2011; Fikar and Hirsch, 2015) simply impose a maximum time limit that a nurse can work without pauses. Finally, several works address settings where there exist services requiring the synchronized presence (SY) of multiple nurses or where there are explicit hard priorities to satisfy (HP). The presence of these constraints highly increase the complexity of the problem and, therefore, the papers studying these variants address instances where the number of services provided and nurses available are much lower than in other works.

The third interesting dimension analyzed in Table 1 concerns the different solution approaches proposed, namely local search methods, approximation algorithms, metaheuristics, matheuristics, and exact methods. Given the computational complexity of several NRP variants, only a few works propose exact methods (Dohn et al., 2009; Trautsamwieser and Hirsch, 2011; Trautsamwieser et al., 2011; Rasmussen et al., 2012; Manerba and Mansini, 2016). Most of the authors develop metaheuristics, possibly hybridized with mathematical programming techniques (matheuristics). Interesting enough, the most frequent approaches are based on the Variable Neighborhood Search framework (see, e.g., Nasir and Dang (2018), dealing with the possibility of selecting new patients and hiring new nurses, and Lasfargeas et al. (2019), studying an NRP with temporal precedences and synchronized services). An Adaptive Variable Neighborhood Search is proposed in Mankowska et al. (2014) where neighborhoods are explored according to a predefined order. Also Cinar et al. (2021) adopt an ALNS based on heuristic neighborhoods to solve a multi-period NRP problem with time windows. Many other metaheuristic paradigms have been used as well, such as Tabu Search (Rest et al., 2012), Multi-Directional Local Search (Braekers et al., 2016), Simulated Annealing (Fathollahi-Fard et al., 2018), or population-based methods like Genetic Algorithms (Yalçindağ et al., 2016), Particle Swarm Optimization (Akjiratikarl et al., 2007), and Fuzzy Simulated Evolution Algorithm (Mutingi and Mbohwa, 2014). Concerning matheuristic approaches, Allaoua et al. (2013) decompose its problem into two MILP formulations, a set partitioning (representing the scheduling part) and a Multi-Depot Traveling Salesman Problem (representing the routing part). Also Fikar and Hirsch (2015) propose a matheuristic method consisting of two stages, the first one to identify feasible routes and the second one to optimize the transportation part. Finally, some Linear Programming techniques are also applied in Bertels and Fahle (2006) and Bredstrom and Ronnqvist (2008).

In conclusion, to the best of our knowledge, no previous work apart from Manerba and Mansini (2016) has addressed the specific NRP variant studied in this paper, which simultaneously considers the working time regulations, the incompatibilities between requests, and a minimum demand for the service types. Moreover, different from the majority of the existing approaches that are based on VRP models, this NRP is formulated as a Team Orienteering Problem (TOP) variant in which the patients to visit and the requests to fulfill must be selected in order to maximize the total profit collected. Finally, despite the fact that we study the demand-constrained NRP variant already proposed in Manerba and Mansini (2016), our compact mathematical formulation and our matheuristic approach (i.e., an ALNS hybridized with a Kernel Search) are new. To date, for the problem at hand, there only exist exact methods that can deal with instances of very small size.

3. Problem definition and formulation

In this section, we present in detail the operational setting in which the NRP-IS is defined, propose an MILP compact formulation for this problem and some valid inequalities to strengthen it.

3.1. Problem statement and notation

Let F be a set of nurses (located at the hospital), let $M = \{1, \dots, m\}$ be the set of patients (spread over a geographical area) and let $K = \{1, \dots, k_{max}\}$ be the set of service types that patients can ask for. We define as $N = \{1, \dots, n\}$ the set of all service requests coming from all the patients. N is partitioned into m nonempty disjoint subsets $N_h, h \in M$, that is, $N = \cup_{h=1}^{m} N_h$ and $N_h \cap N_{h'} = \emptyset$ for $h \neq h'$, where each N_h represents the subset of service requests coming from patient $h \in M$. For each request $i \in N$, $\sigma(i) : N \mapsto K$ is a function that associates a service type in K with each request i, st_i represents the time needed to perform the service, while p_i is the profit (priority) assigned to such a request. We assume that each nurse has the skills to perform any service type and thus can serve any patient. Moreover, she can move with her own vehicle. Due to the service provider's internal policies and possible negative interactions among services, incompatibilities among some service types are established. We define B as the (possibly void) set of pairs of service types that are incompatible and, in turn, as $\bar{B} = \{[i, j] : i \neq j \in N_h, h \in M, [\sigma(i), \sigma(j)] \in B\}$ as the set of pairs of requests that cannot be both performed to a patient.

Let us define a directed graph $G = (V, A)$ with node set $V = N \cup \{0, n+1\}$, where nodes 0 and $n+1$ are the starting and ending hospitals (possibly the same), and arc set $A = \{(i, j) : i, j \in V, i \neq j, [i, j] \notin \bar{B}\}$. A nonnegative traveling time t_{ij} is assigned to each arch $(i, j) \in A$, where $t_{ij} = 0$ if nodes i and j belong to the same $N_h, h \in M$ (i.e., i and j are requests coming from the same patient h). Traveling times are assumed to satisfy the triangle inequality. Every morning, all the nurses leave the hospital (node 0) at time 0. Each of them has a maximum working time equal to t_{max}, considering both service and traveling times. We assume $st_0 = st_{n+1} = 0$ and $p_0 = p_{n+1} = 0$. The NRP-IS aims at determining, for each nurse, the sequence of requests to perform in a single visiting tour (i.e., to determine which patients to visit and which services to fulfill for each patient) so to maximize the overall profit (priority) given by the caring operations while complying with the daily working time t_{max} for each nurse, and all the incompatibilities among service requests. Note that satisfying all the requests of a patient is not a requirement. More precisely, since in general the requests are too many to be accomplished by the available nurses in a day, the problem does not force to fulfill all the requests, neither to visit all the patients nor to perform all the services required by a patient if visited. This means that, if convenient, more than one nurse can visit the same patient, each one performing a different service.

Finally, we also model the generalization NRP-ISD where a minimum number d_k of service requests needs to be guaranteed for each $k \in K$. The rationale is to guarantee a fairer global service by avoiding that only high profitable services are performed.

3.2. A new compact two-index formulation

Different mathematical formulations, based on a graph where each node is associated with a patient instead of with a service request, have been proposed for the problem in Manerba and Mansini (2016). In the following, we propose a new compact one.

Let us introduce a binary variable w_i, $\forall i \in N$, taking value 1 if request i is satisfied, and 0 otherwise, a binary variable x_{ij}, $\forall (i, j) \in A$, taking value 1 if arc (i, j) is traversed, and 0 otherwise, and a continuous variable q_{ij} that, $\forall (i, j) \in A$, measures the time of arrival at node j coming from node

i. Moreover, we denote as $\delta^+(\bar{V})(\delta^-(\bar{V}))$, for a given set $\bar{V} \subset V$, the set of arcs $(i, j) \in A$ with $i \in \bar{V}$ and $j \in V \setminus \bar{V}$ ($i \in V \setminus \bar{V}$ and $j \in \bar{V}$). Then, the proposed mathematical formulation is as follows:

$$\max \quad \sum_{i \in N} p_i w_i, \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in N : \sigma(i) = k} w_i \geq d_k \qquad k \in K, \tag{2}$$

$$\sum_{(j,i) \in \delta^-(\{i\})} x_{ji} = \sum_{(i,j) \in \delta^+(\{i\})} x_{ij} = w_i \qquad i \in N, \tag{3}$$

$$\sum_{(0,j) \in \delta^+(\{0\})} x_{0j} = \sum_{(i,n+1) \in \delta^-(\{n+1\})} x_{i,n+1} \leq |F|, \tag{4}$$

$$\sum_{(i,j) \in \delta^+(\{i\})} q_{ij} - \sum_{(j,i) \in \delta^-(\{i\})} q_{ji} = \sum_{(i,j) \in \delta^+(\{i\})} (t_{ij} + st_i) x_{ij} \qquad i \in N, \tag{5}$$

$$q_{0i} = t_{0i} x_{0i} \quad i \in N, \tag{6}$$

$$(t_{0i} + t_{ij} + st_i) x_{ij} \leq q_{ij} \leq (t_{max} - st_j - t_{j,n+1}) x_{ij} \quad (i, j) \in A, i \neq 0, \tag{7}$$

$$w_i + w_j \leq 1 \qquad [i, j] \in \bar{B}, \tag{8}$$

$$w_i \in \{0, 1\} \qquad i \in N, \tag{9}$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in A, \tag{10}$$

$$q_{ij} \geq 0 \qquad (i, j) \in A \setminus \{(0, n+1)\}. \tag{11}$$

The objective function (1) aims at maximizing the sum of profits obtained by satisfying the patient requests. Constraints (2) ensure that at least d_k requests are performed for each type of service $k \in K$. Constraints (3) are classical pairing constraints, whereas constraints (4) control the creation of at most $|F|$ tours. Constraints (5) guarantee that if node j is visited after node i (i.e., request j is executed after request i), then the time elapsed between the arrival times in the two nodes (i.e., the starting times for the execution of the services) has to be equal to the service time st_i plus the traveling time spent to move from i to j. Constraint (6) set the value of variables q_{0i} equal to the traveling time to reach node i from node 0, if it is visited. Constraints (7) define upper and lower bounds for each variable q_{ij} if arc $(i, j) \in A$ is traversed. Note that, even if the lower bound is not necessary being indirectly implied by constraints (5) and (6), we introduce it as it helps strengthening the LP relaxation. Moreover, note that constraints (5)–(7) ensure, in a compact way, the elimination of subtours (see Maffioli and Schiomachen, 1997). Constraint (8) impose that at most one service request in the pair $[i, j]$ can be satisfied if such requests are incompatible. Finally, the remaining constraints (9)–(11) are binary and nonnegative conditions on variables.

Note that constraints (2) have been introduced to model the problem with demand (NRP-ISD). When $d_k = 0, \forall k \in K$, constraints (2) become useless and the model boils down to the NRP-IS.

3.3. Valid inequalities

The model can be strengthened by the introduction of some valid inequalities.

First, similarly to Bianchessi et al. (2018), we impose an upper bound to the global traveling time spent by all the nurses by using

$$\sum_{(i,j)\in A} (t_{ij} + st_j)x_{ij} \leq |F|\, t_{max}. \tag{12}$$

Second, even though constraints (5)–(7) already ensure the subtours elimination, we also include the so-called General Connectivity Constraints (GCCs) expressed by

$$\sum_{(i,j)\in \delta^+(S)} x_{ij} \geq w_l, \quad S \subset N, |S| \geq 2, l \in S. \tag{13}$$

Third, since each patient cannot be visited by the same nurse more than once, if several requests are accomplished for the same patient by a nurse, then she has to achieve them all in a row, that is

$$\sum_{i\in N_h:(j,i)\in A} x_{ji} \leq 1 - x_{lj}, \quad h \in M, l \in N_h, j \in N \setminus N_h : (l, j) \in A. \tag{14}$$

The inequality considers a certain request l from a patient h and, if such a request is accomplished and a request j from another patient is served immediately after, then all the arc variables that would allow the nurse to come back from j to patient h are set to 0.

Finally, note that when an arc between two nodes i and j has been selected (either $x_{ij} = 1$ or $x_{ji} = 1$), then $w_i = w_j$. This can be imposed by the following Arc-Vertex Inference Cuts inequalities introduced by Assunção and Mateus (2021):

$$\begin{aligned} w_i - w_j &\leq 1 - (x_{ij} + x_{ji}) \quad (i, j) \in A, i \neq 0, j \neq n+1, \\ w_j - w_i &\leq 1 - (x_{ij} + x_{ji}) \quad (i, j) \in A, i \neq 0, j \neq n+1. \end{aligned} \tag{15}$$

4. The hybrid ALNS solution framework

ALNS is a metaheuristic framework (Ropke and Pisinger, 2006, Pisinger and Ropke, 2007) based on the *Ruin and Recreate* paradigm (Schrimpf et al., 2000) and extending the Large Neighborhood Search (LNS) presented by Shaw (1998). At each iteration, ALNS selects one operator to destroy the current solution and another one to repair it. The choice is made among predefined heuristic operators in an adaptive way, that is, the higher the quality of the solutions found using an operator, the higher the probability the same operator is selected again in the subsequent iterations. After a certain number of iterations (an *epoch*), such probabilities are reset. In our problem,

Algorithm 1. `ALNS-KS`

```
Require: a solution $s_I$ with value $obj_{s_I}$; the sets $\Omega^-$ and $\Omega^+$ of destroy and repair operators; a maximum execution
  time $\tau$.
 1: $s_C \leftarrow s_I$ ▷ current solution
 2: $s_B \leftarrow s_I$ ▷ best incumbent solution
 3: $q \leftarrow 1$
 4: repeat
 5:   choose $\omega^- \in \Omega^-$ and $\omega^+ \in \Omega^+$
 6:   $[\bar{s}_C, N^-] \leftarrow \omega^-(s_C, q)$
 7:   $s_E \leftarrow \omega^+(\bar{s}_C, N^-)$
 8:   if $s_E$ is accepted then
 9:     $s_C \leftarrow s_E$
10:     if $obj_{s_E} > obj_{s_B}$ then
11:       $s_B \leftarrow s_E$
12:     end if
13:     $q \leftarrow 1$
14:   else if no solution accepted for $thrImp$ then
15:     $q \leftarrow q + qStep$
16:   end if
17:   update probabilities of $\omega^-$ and $\omega^+$
18:   if $maxIterPerEp$ is reached then
19:     reset probabilities of all operators in $\Omega^-$ and $\Omega^+$
20:     $s_C \leftarrow s_B$
21:     if $s_B$ not improved for $thrEpImp$ then
22:       $s_C, s_B \leftarrow$ KS()
23:     end if
24:   end if
25: until time limit $\tau$ is reached
```

a generic solution s (with a value indicated by obj_s) is composed by a set of $|F|$ routes, one for each nurse, represented as sequences of service requests. All the destroy operators remove, according to a specific criterion, a subset of service requests. The resulting partial solution is repaired by using different operators that try to insert nodes into routes following a specific order. Actually, we propose a hybrid ALNS algorithm exploiting the exact resolution of MILP models (see, e.g., Grangier et al., 2017 or Mansini and Zanotti, 2020). In particular, a Kernel Search approach has been devised as an intensification procedure and comes into play when the solution quality improvement is stalling during the ALNS execution. Given the hybrid nature of the developed approach, we call the resulting matheuristic `ALNS-KS`.

The general structure of `ALNS-KS` is reported in Algorithm 1. The algorithm requires an initial feasible solution s_I (found by using the greedy procedure described in Section 4.1), sets Ω^- and Ω^+ containing the destroy and repair operators, and the maximum execution time τ. In Steps 1 and 2, the current solution s_C and the best incumbent integer solution s_B are initialized to the initial solution s_I. Moreover, in Step 3, the parameter q representing the *degree of destruction* (i.e., the number of requests removed from the solution) is initialized to 1. The main loop (Steps 4–25) starts and is repeated until the elapsed time does not exceed the threshold time τ. In Step 5, a destroy operator ω^- and a repair operator ω^+ are chosen. Then, in Step 6, operator ω^- partially destroys the current solution s_C according to the degree of destruction q, thus generating a partial solution $\bar{s}_C$, together with the set $N^- \subset N$ containing all the requests removed from s_C (see Section 4.2). Then, in Step 7, ω^+ repairs the partial solution $\bar{s}_C$ without using the just removed requests in N^-, thus generating a potentially improved solution s_E (see Section 4.3). If s_E is accepted, which occurs with a probability calculated according to the classical Simulated Annealing-like rule used in Pisinger and Ropke (2007), the current solution and possibly (in case of improvement) the best solution are updated (Steps 9–12), and the parameter q is reset to the initial value 1 (Step 13). Moreover, if the number of iterations since the last accepted solution has reached a certain threshold $thrImp$, then q is incremented by a value $qStep$ (Steps 14 and 15). In Step 17, the probabilities of the two chosen operators are updated. We precise that, at the first iteration, each operator has the same probability to be chosen. During the subsequent iterations, as originally proposed in Ropke and Pisinger (2006), we update the probability for each operator according to its performance, that is, we increase its probability if it has produced an accepted solution and decrease it otherwise. Finally, Steps 18–24 provide intensification routines. In particular, when an epoch has passed, that is, every $maxIterPerEp$ iterations, the selection probabilities of all destroy and repair operator are reset to the initial value and the search restarts from the best solution obtained so far s_B. Moreover, if a predefined number $thrEpImp$ of epochs passed without an improvement of the best solution, we call a Kernel Search that attempts to improve the current best solution by solving a sequence of restricted integer problems built on the initial formulation NRP-IS by considering only the arcs belonging to the solutions visited since the last improvement or the last KS run (see Section 5).

4.1. Heuristic algorithms for finding the initial solution

The initial solution is constructed in a different way according to the variant of the problem.

For the NRP-IS, the method starts from an empty solution containing nodes 0 and $n + 1$ in all routes, processes all requests in random order, and tries to insert each one according to a cheapest insertion policy. If a request cannot be inserted in any route due to incompatibilities or workload time restriction, it is discarded. Classical *2-opt* and inter-route *node-swap* procedures are performed after each insertion. This procedure is reiterated multiple times until a stopping condition is met (either n_{init} solutions have been produced or a time limit τ_{init} has been reached). The initial solution s_I given to ALNS-KS is the one with the highest profit among the ones identified.

For the NRP-ISD, instead, we use a two-phase approach that focuses on satisfying as soon as possible the minimum demand for the service types. In the first phase, we proceed as in the previous case, but we consider only those requests associated with a service type that has a nonnull minimum demand. Once the demand for a certain service type has been totally satisfied, we discard the requests not included in the solution and associated with such a service type. If the above procedure does not produce a feasible solution, it restarts from scratch. In the second phase, we proceed exactly as in the case without demand, considering in random order all the requests that have not been already included so far in the solution.

Please note that, while the procedure described above for NRP-IS guarantees to find a feasible solution, this is not always true regarding the NRP-ISD variant. In the unlucky cases yielding infeasibility, we can use an MILP solver to find a feasible assignment of the nurses to the requests by solving a simplified version of model (1)–(11) that considers only those requests with a positive minimum demand. However, in our computational tests, we never experienced such pathological situations.

4.2. Destroy operators

Each destroy procedure concerns the removal of different requests from a current solution. We call N^- the set of requests to remove. This set is created by sorting in a specific order the requests belonging to s_C and by selecting, one by one with the 80% of probability, a number of requests equal to q. We implemented six different sorting procedures, thus yielding six corresponding destroy operators:

- ω_1^-: the requests are sorted randomly;
- ω_2^-: the requests are sorted in nondecreasing order of profit over the sum of the travel times of the corresponding incoming and outgoing arcs;
- ω_3^-: the requests are sorted in nondecreasing order of profit over service time;
- ω_4^-: the requests are sorted in nondecreasing order of profit over the sum of the travel times of the corresponding incoming and outgoing arcs plus the service time;
- ω_5^-: the requests are sorted in nondecreasing order of profit over the maximum profit of all the requests incompatible with the considered one (if the request does not have any incompatibilities, only the profit is used);
- ω_6^-: the requests are sorted in nonincreasing order of the percentage of solutions in which such a request appears since the last best solution improvement (or the last KS run).

Additionally, we implement one last destroy operator ω_7^- based on the well-known *Shaw's removal* concept. In this operator, one request is initially selected randomly from s_C and added to N^-. Then, until $|N^-| < q$, the algorithm first selects randomly a request from N^- and then adds to N^- the most *similar* one from s_C and not yet included in N^-. The similarity between two requests i and j is determined through a *relatedness measure*$R(i, j) = 1/(t_{ij} + 0.1|st_i - st_j| + \mathcal{I}_{ij})$, where $\mathcal{I}_{ij} = 1$ if the two requests are served by the same nurse, and 0 otherwise.

4.3. Repair operators

The repairing procedure aims at reconstructing the current partial solution by trying to add to its requests those belonging to a set $N^+ = N \setminus N^-$. The procedure sorts in a specific order the requests in N^+ and tries to insert them one by one with the 80% of probability if the feasibility is maintained. Note that, for the NRP-ISD variant, the sorting gives priority to requests associated with a nonnull demand service type, until the demand is not satisfied.

We implemented four different sorting rules, mirroring those proposed in the first four destroy operators:

- sr_1: the requests are sorted randomly;

- sr_2: the requests are sorted in nonincreasing order of profit over the sum of the travel times from the two requests in solution closest to the considered one;
- sr_3: the requests are sorted in nonincreasing order of profit over service time;
- sr_4: the requests are sorted in nonincreasing order of profit over the sum of the travel times from the two requests in solution closest to the considered one plus the service time.

For the last three sorting rules, we also implemented two *regret*-based versions. Here, we compute for the first $2|N^-|$ requests a regret score and we re-sort them accordingly in nonincreasing order. For each request, the *regret-2* score is calculated as the difference between the cost of inserting it in the second-best route by using a cheapest insertion algorithm and that of doing the same for the very best route. Instead, the *regret-3* score is calculated as the *regret-2* score plus the difference between the cost of inserting the request in the third-best route and that of doing the same for the very best one. Eventually, 10 repair operators arise:

- ω_1^+: the *no-regret* repair operator based on the sorting rule sr_1;
- $\omega_2^+, \omega_3^+, \omega_4^+$: the *no-regret* repair operator based on sorting rule sr_2, sr_3, sr_4, respectively;
- $\omega_5^+, \omega_6^+, \omega_7^+$: the *regret-2* repair operator based on sorting rule sr_2, sr_3, sr_4, respectively;
- $\omega_8^+, \omega_9^+, \omega_{10}^+$: the *regret-3* repair operator based on sorting rule sr_2, sr_3, sr_4, respectively.

Note that, only in the NRP-ISD case, the definition of N^+ does not ensure to obtain a repaired feasible solution because the requests necessary to achieve the minimum demand for a service could have been excluded by the destroy operator used. In this case, the algorithm simply stops the repairing procedure.

5. Kernel Search-based intensification procedure

In this section, we present our intensification method based on the Kernel Search. We first give a brief overview of the method, then we describe how it is implemented and embedded into our framework.

5.1. Kernel search

Kernel Search is a well-known heuristic framework initially proposed for the solution of general MILP problems and successfully applied to many specific problems such as knapsack problems (Angelelli et al., 2010; Lamanna et al., 2022), portfolio selection (Angelelli et al., 2012), and routing problems (Hanafi et al., 2020).

The method is based on the construction of a sequence of restricted problems solved by means of an MILP solver (Gurobi, Cplex). Each restricted problem includes only a subset of variables of the original problem, whereas the remaining ones are set to zero. To built restricted problems, the algorithm identifies the most promising variables. More precisely, all variables are sorted according to a predefined rule so that more promising ones come first. A variable is highly promising if it is highly probable it will be selected in an integer optimal solution. Even if several methods exist in the literature, a classical way to derive this probability is to use the value of variables belonging to the basis in the optimal solution of the continuous relaxation and the absolute value of the reduced cost for the out-of-the-basis ones. The first variables in the ordered list enter the so-called *kernel set*, the remaining ones are partitioned into groups called *buckets*. The first restricted problem contains the variables of the initial kernel set, each one of the others is constructed by jointly considering the variables in the kernel set plus those belonging to one bucket. The construction of the kernel set is not a one-shot procedure but is based on a learn-and-adjust method where new variables are added and some others are excluded at each iteration when solving a new restricted problem. The idea is that the final kernel set should contain most of (hopefully all) the variables that will be selected in an optimal solution. Note that the number of restricted problems to solve depends on the number of constructed buckets and is a parameter of the method. Either a few or all possible buckets can be considered. It is even possible to scroll buckets more than once. Finally, since optimally solving the restricted problems could be computationally too cumbersome, a solution time limit can be imposed. If this is the case, for each restricted problem, the solver will possibly provide the best feasible solution found so far instead of the proven optimal one.

5.2. Implementation details

In our intensification procedure, we implemented a slightly different KS. An innovative aspect is that, differently from the existing works on KS, promising variables are identified not by means of the LP relaxation but exploiting the information coming from a pool of solutions generated by another algorithm execution (the ALNS in our case). More precisely, in the NRP-IS formulation, we have three types of variables: those associated with patient requests and used to model conflicts (variables w representing nodes of the graph) and two groups of variables associated with connections among requests (variables x representing the arcs selection and variables q associated with arrival times). Instead of sorting all variables, one can note that it is possible to identify a key set of variables controlling all the remaining ones and only sorting this set to create both the initial kernel set and the buckets. Hence, the key set of variables is represented by arc variables x. In fact, a q-variable is active (i.e., not set to 0) if the corresponding x-variable is active, while a w-variable associated with a node is active if at least two arc variables (one entering and one leaving) insisting on such a node are both active.

The ALNS and the KS methods cooperate as follows. Let us define as S_{ALNS} the set of solutions produced by `ALNS-KS` since the last global improvement (or the last KS run) sorted in nonascending order of value (if a tie occurs, the solution with the lower amount of nurses working time comes first). For each arc, we compute a score depending on the quality of the solutions in S_{ALNS} in which it is selected. More precisely, each time an arc appears in the rth solution of the sorted list, its score gains a value equal to $|S_{ALNS}| - r$. The kernel set is constructed by adding (i) all the x variables corresponding to the arcs selected in the best incumbent integer solution identified so far by `ALNS-KS`; (ii) two times the number of variables in (i) selected among those with the highest scores. The remaining variables, sorted by score, are divided into four buckets of fixed size equal to 500. We decided to consider only this limited number of buckets since the KS is assumed to be called with a very tight time budget.

The time granted to KS is 15 seconds and it is equally divided among the five restricted problems to solve. Note that, if some time remains after the solution of the last restricted problem and at least an improved solution has been found, then the KS continues to iterate on solving the restricted problems including the updated kernel set.

We also precise that the mathematical formulation used by KS is the one exposed in (2)–(11) with the following slightly modified objective function:

$$\max \quad \sum_{i\in N} p_i w_i - \alpha \sum_{i\in N} q_{i,n+1}. \tag{16}$$

Equation (16) adds to the total profit a penalty that depends on the total working time of the nurses, scaled by a parameter $\alpha = 0.001$. This change allows the KS to possibly improve a solution both because it achieves a higher profit and because it reduces the total time needed to serve the requests.

As mentioned, KS is used as an intensification procedure around the best solution when no improvements have occurred for several ALNS iterations. Since KS is based on the solutions collected during the ALNS execution, its effectiveness strongly depends on the number of solutions that are close in value to the best one but different in terms of arcs and nodes selected. That said, in order to obtain a set S_{ALNS} with a consistent number of high-quality solutions, KS is invoked only if the best solution has not been improved for a predefined threshold *thrEpImp* of consecutive epochs (see Steps 21–23 of Algorithm 1). This way, in our experiments S_{ALNS} has usually resulted in containing several thousands of solutions.

Finally, to allow KS to obtain some improvements, even in the case of a high quality incumbent solution (as it usually happens in the second-half of the ALNS execution), we have increased the KS time limit by 10% if the previous KS has not produced any improvement and no improvement has occurred since the last KS run. Note that, the time limit is reset any time a global improvement is achieved.

6. Computational experiments

This section is devoted to describe the computational results obtained by testing the proposed mathematical formulation and the implemented `ALNS-KS` and its components. We discuss the generation of the instances in Section 6.1, provide some implementation details of our algorithms in Section 6.2, and comment on the results obtained in Section 6.3. All the computational tests have been run on an *AMD Ryzen 9 3950x* machine using four of its cores and 32GB of RAM, and running a 64-bit *Windows 10* operating system. The MILP solver used is Gurobi v9.1.2 and all the algorithms have been implemented in Java.

6.1. Instances generation

To create realistic scenarios for our computational tests, we generated instances in which each nurse works for $t_{max} = 360$ minutes (six hours) per day, the patients and the hospital are geographically dispersed over a 30×30 km^2 square area (which is the typical covering area of an Italian healthcare provider), and the average traveling speed of nurses is 60 km/h. Traveling times are simply computed as Euclidean distances between locations divided by such an average speed.

Table 2
Possible parameter combinations for instance generation

$\|F\|$	$\|M\|$	$\|K\|$	$\|N\|$	λ%
4	50	5	100, 125	25, 50
5	50	10	150, 175	25, 50
6	50	5	200, 250	25, 50
7	50	10	300, 350	25, 50

Each instance involves up to 60% of short-duration services (5–15 minutes), 40% of medium-duration services (15–45 minutes), and 20% of long-duration services (45–95 minutes). The first category includes, for example, the administration of various drugs, the measurement of vital signs, and blood or other organic material tests. The second category includes ordinary or post-operative bandages, dressings, and injections. The more complex tests and therapies, such as electrocardiogram, dialysis, chemotherapy, and physiotherapy sessions, fall into the last category. We consider scenarios in which the hospital provides no more than 10 types of services and at most λ% of these services are involved in an incompatibility. Typical incompatibilities arise, for example, between the conjunct administration of medicines that could create negative side effects, or between drugs that could induce excessive bleeding and post-operative medications. We assume that each patient requests at least 10% and at most 70% of the available services, and for each request i, a profit p_i is randomly generated in [1, 200]. Eventually, we generate 80 NRP-IS instances, that is, five random repetitions for each one of the 16 combinations of number of nurses $|F|$, patients $|M|$, service types $|K|$, total requests $|N|$, and value of λ% presented in Table 2.

Finally, in order to address also the NRP-ISD variant, for each one of the previous 80 instances, a minimum demand d_k for each service type $k \in K$ is generated randomly in $[0, |\{i \in N : \sigma(i) = k\}|/2]$. This yields 160 instances in total. All the generated instances, along with detailed results of the tests described in Section 6.3, are available at the webpage https://or-dii.unibs.it/index.php?page=nrpis.

6.2. Implementation details and parameters tuning

Hereafter, we call `EXACT` our branch-and-cut method embedding the resolution of the mathematical model (1)–(11) through the Gurobi MILP solver, with a time limit of 3600 seconds, and the addition of valid inequalities described in Section 3.3. In particular, valid inequality (12) is added from scratch. Instead, the other proposed valid inequalities are dynamically separated in the branch-and-bound's tree by using Gurobi's callbacks. The separation of GCCs in (13) can be done in polynomial time by using consolidated max-flow based techniques (see, e.g., Beraldi et al., 2017). We solve the involved max-flow problems by using the algorithm proposed in Boykov and Kolmogorov (2004). Instead, the searching for violated inequalities (14) and (15), which are polynomial in number, is simply done by enumeration. Finally, in Table 3, we summarize all the parameters used for the `ALNS-KS` execution and provide their value. The tuning has been done over a representative subset of instances.

Table 3
Tuning of the `ALNS-KS` parameters

Parameter	Meaning	Value
$qStep$	Incremental quantity of the degree of destruction q	1
$thrImp$	Number of iterations without improvement before increasing q	100
$maxIterPerEp$	Number of iterations per epoch	3000
$thrEpImp$	Number of epochs without improvement after which KS is run	5
n_{init}	Number of solutions produced by the initial solution heuristics	200
τ_{init}	Time limit for the initial solution heuristics	20 s

Table 4
`EXACT` versus `MM16`

		`EXACT`				`MM16`				
Variant	Stat.	ttb	$\#bbn$	$gap(\%)$	$\#f$	ttb	$\#bbn$	$gap(\%)$	$\#f$	$\Delta_{obj}(\%)$
NRP-IS	Avg	2436	6250	5.61	16	1324	40,130	62.72	16	−60.30
	Min	418	3	2.41		0	6580	6.83		−100.00
	Max	3534	30,608	16.29		3527	86,891	100.00		−0.29
NRP-ISD	Avg	3056	5856	7.87	9	2556	101,514	8.73	1	−3.16
	Min	2008	1	1.93		2556	343,517	8.73		−3.16
	Max	3601	34,373	21.45		2556	23,041	8.73		−3.16

6.3. Results and discussion

In this section, we report the computational experiments performed on both the problem variants (NRP-IS and NRP-ISD) and discuss the results obtained by our algorithms. In Section 6.3.1, we evaluate the effectiveness of our new model and of the valid inequalities introduced. In Section 6.3.2, we analyze the performance of our `ALNS-KS` in terms of efficiency and quality of the solutions, while in Section 6.3.3 we assess the contribution of the KS intensification.

6.3.1. Mathematical formulations evaluation

In the following preliminary experiments, we first assess our branch-and-cut (`EXACT`) with respect to the one already available in the literature (proposed in Manerba and Mansini, 2016 and named `MM16` hereafter) and based on a different model. Table 4 shows this comparison by reporting, for NRP-IS and NRP-ISD, some statistics on a subset of 32 instances (one for each variant and for each parameter combination presented in Table 2). No runs have reached the optimality within one hour. For each method, the columns report:

- ttb: the time-to-best in seconds, that is, the time at which the best solution has been found;
- $\#bbn$: the number of branch-and-bound nodes explored during the procedure;
- gap: the percentage gap between the best feasible solution lb and the best upper bound ub found by the relative method. It is calculated as $100\frac{ub-lb}{ub}$;
- $\#f$: the number of instances for which at least a feasible solution has been found.

Table 5
Valid inequalities contribution

		`EXACT`							`EXACT-noCuts`						
Variant	Stat.	*ttb*	# *bbn*	#(13)	#(14)	#(15)	*gap*(%)	#*f*	*ttb*	#*bbn*	*gap*(%)	#*f*	Δ_{obj}(%)	Δ_I(%)	Δ_F(%)
NRP-IS	Avg	2400	5744	10,818	720	1011	8.83	32	1996	6066	9.48	32	−0.65	−0.45	−0.42
	Min	992	21	1456	70	294	2.29		234	4	2.36		−4.25	−0.15	−0.67
	Max	3280	33,476	39,772	3987	4063	13.63		3489	34,098	14.03		1.68	−0.65	−0.01
NRP-ISD	Avg	2746	6357	17,107	652	1565	7.32	15	2768	98,37	7.12	14	0.43	−0.88	−0.55
	Min	1875	52	5224	73	376	1.94		1741	1	2.66		−3.59	−2.20	−1.29
	Max	3582	35,903	56,006	4234	7239	21.45		3458	54,650	19.95		4.01	−0.33	−0.06

Finally, Δ_{obj} indicates the percentage gap between $lb_{\texttt{EXACT}}$ (the best solution found by `EXACT`, if any) and the corresponding value $lb_{\texttt{MM16}}$ for `MM16`, calculated as $100\frac{lb_{\texttt{MM16}}-lb_{\texttt{EXACT}}}{lb_{\texttt{EXACT}}}$. Clearly, a negative value means that `EXACT` has found a better solution than `MM16`.

From Table 4, it clearly appears that our new branch-and-cut totally outperforms the existing method. In the NRP-IS instances, for which `MM16` is still able to always find at least a feasible solution, `EXACT` achieves solutions with an average improvement in objective function of 60% and with an average *gap* that is about 10 times lower. Moreover, note that the number of branch-and-bound nodes explored is about 8 times lower. This is a clear consequence of the addition of the valid inequalities. When considering NRP-ISD instances, instead, even our branch-and-cut starts getting less effective. Here, a direct comparison between the two methods is biased by the fact that `MM16` is able to find a feasible solution only in one case. However, `EXACT` finds a feasible solution in 9 out of 16 cases and strongly reduces the average number of branch-and-bound nodes explored.

To complete the analysis, we assess the effectiveness of the valid inequalities presented in Section 3.2 by comparing `EXACT` with its version without cuts separation (`EXACT-noCuts` hereafter). Table 5 shows this comparison by reporting, for NRP-IS and NRP-ISD, some statistics on a subset of 64 instances (two repetitions for each variant and for each parameter combination presented in Table 2). Again, no runs have reached the optimality within one hour. In addition to the column headers already explained,

- #(13), #(14), and #(15) are, for each method, the number of violated valid inequalities (13)–(15) added, respectively;
- Δ_{obj} now indicates the percentage gap between $lb_{\texttt{EXACT}}$ and $lb_{\texttt{EXACT-noCuts}}$, that is, the best solution found (if any) by `EXACT` and `EXACT-noCuts`, respectively. It is calculated as $100\frac{lb_{\texttt{EXACT-noCuts}}-lb_{\texttt{EXACT}}}{lb_{\texttt{EXACT}}}$;
- Δ_I is the percentage gap between the initial upper bound (just after the root node) for `EXACT` and `EXACT-noCuts`, that is, $ub^i_{\texttt{EXACT}}$ and $ub^i_{\texttt{EXACT-noCuts}}$, respectively. It is calculated as $100\frac{ub^i_{\texttt{EXACT}}-ub^i_{\texttt{EXACT-noCuts}}}{ub^i}$;
- Δ_F is the percentage gap between the final upper bound for `EXACT` and `EXACT-noCuts`, i.e. $ub^f_{\texttt{EXACT}}$ and $ub^f_{\texttt{EXACT-noCuts}}$, respectively. It is calculated as $100\frac{ub^f_{\texttt{EXACT}}-ub^f_{\texttt{EXACT-noCuts}}}{ub^f}$.

Concerning the NRP-IS variant, more than 12,000 valid inequalities are added on average: cuts (13) are the 85%, while inequalities (14) and (15) are much less, about 6% and 9% of the total, respectively. The *gap* columns show that for EXACT the *gap* drops by about 0.7%, compared to EXACT-noCuts, while on average the objective function value improves by 0.65% (with some peaks beyond 4%). From the Δ_I and Δ_F columns, it is clear that the introduction of valid inequalities allows to improve both the initial and final upper bounds, by 0.45% and 0.42%, respectively. Instead, concerning the NRP-ISD variant, on average about 20,000 valid inequalities are added in each resolution: 89% of (13), 8% of (15), the remaining 3% of (14). This great number of added cuts does not allow to clearly assess if EXACT outperforms EXACT-noCuts in terms of objective function (slightly worse average, and oscillating min and max values). However, the efficiency of the valid inequalities is evident also for the NRP-ISD from the columns Δ_I e Δ_F, where the initial and final upper bounds are on average improved by 0.88% and 0.55%. Finally, note that EXACT and EXACT-noCuts do not manage to find a feasible solution within the time limit for 17 and 18 instances, respectively.

In conclusion, we definitely decided to use EXACT any time an MILP restricted problem solution is called inside ALNS-KS.

6.3.2. *Evaluation of the* ALNS-KS *performance*

In the following, we assess the performance of the heuristic framework ALNS-KS with respect to EXACT. Table 6 reports this comparison by showing in the columns, for both problem variants and for $\tau = 1, 5$, and 10 minutes:

- Δ^a_{obj}, Δ^b_{obj}, Δ^w_{obj}: average, best, and worst values over five runs of Δ_{obj}, now representing the percentage gap between the best solution found (if any) by EXACT ($lb_{\texttt{EXACT}}$) and the solution found by ALNS-KS ($lb_{\texttt{ALNS-KS}}$), calculated as $= 100\frac{lb_{\texttt{EXACT}}-lb_{\texttt{ALNS-KS}}}{lb_{\texttt{EXACT}}}$;
- *ttb*: time-to-best in seconds, that is, the time at which the best solution has been found by ALNS-KS;
- *#it*: thousands of ALNS iterations;
- *#KS*: number of Kernel Search calls.

Each row shows average results per number of requests. A dash indicates that $lb_{\texttt{EXACT}}$ is not available.

At this disaggregated level, we just highlight that our ALNS-KS shows a very stable behaviour over the five runs. For a more compact analysis, Figs. 1 and 2 show, for NRP-IS and NRP-ISD, respectively, the boxplot statistics of Δ^{avg}_{obj}. The data are reported for the three different τ values and aggregated for number of requests. In general, we can see that a longer time does not significantly improve the Δ^{avg}_{obj} in a given subset of instances. However, regardless of the time limit imposed, the gap improves more and more as the number of requests increases. This means that ALNS-KS is able to solve even the most difficult scenarios, where EXACT does not perform well. Concerning NRP-IS, the gap varies between $+1.00\%$ and -1.59% for $|N| = \{100, 125\}$ instances, between -0.01% and -6.62% for $|N| = \{150, 175\}$, between -0.18% and -9.68% for $|N| = \{200, 250\}$, and between -3.20% and -16.91% for $|N| = \{300, 350\}$. The improving trend is very similar for the NRP-ISD but with a bigger proportion, which allows us to obtain even more than 40% of gap in the larger instances. It is worth noticing that, even within 1 minute, ALNS-KS is able to always obtain a better result for instances with more than 150 requests. For the sake of completeness, Fig. 3 shows the percentage of the total number of runs in which ALNS-KS obtains better results than the exact approach. The data are reported separately for each variant and for three different τ values, and aggregated for number of requests. The number of times that ALNS-KS manages to find a better

Table 6
`ALNS-KS` versus `EXACT`

		$\tau = 1$ minute						$\tau = 5$ minutes						$\tau = 10$ min					
$\|N\|$	Variant	Δ^a_{obj}	Δ^b_{obj}	Δ^w_{obj}	*ttb*	*#it*	*#KS*	Δ^a_{obj}	Δ^b_{obj}	Δ^w_{obj}	*ttb*	*#it*	*#KS*	Δ^a_{obj}	Δ^b_{obj}	Δ^w_{obj}	*ttb*	*#it*	*#KS*
100	NRP-IS	−0.3	−0.6	−0.1	26.8	64.9	1.7	−0.6	−0.7	−0.4	139.5	249.1	11.4	−0.7	−0.8	−0.5	294.1	474.9	24.3
	NRP-ISD	−0.5	−0.8	−0.3	33.2	75.6	1.6	−0.9	−1.1	−0.5	182.3	276.5	12.6	−1.0	−1.2	−0.8	310.0	526.2	27.0
125	NRP-IS	−0.4	−0.7	−0.1	28.9	70.7	1.2	−0.7	−1.0	−0.5	162.9	250.3	10.3	−0.9	−1.2	−0.6	330.8	465.5	22.5
	NRP-ISD	0.0	−0.5	0.5	36.0	74.7	1.7	−0.4	−0.8	0.0	191.3	284.6	12.6	−0.6	−0.9	−0.1	341.6	538.6	27.3
150	NRP-IS	−2.0	−2.3	−1.7	27.3	62.4	0.8	−2.3	−2.6	−2.1	171.5	213.2	8.2	−2.4	−2.7	−2.2	339.7	391.1	18.3
	NRP-ISD	−1.8	−2.2	−1.4	33.0	64.3	0.9	−2.3	−2.7	−1.9	205.7	224.8	9.0	−2.5	−2.9	−2.0	404.5	418.8	20.0
175	NRP-IS	−1.2	−1.5	−0.8	34.2	66.7	0.9	−1.5	−1.8	−1.2	162.9	229.2	9.0	−1.6	−1.8	−1.4	330.3	424.8	20.1
	NRP-ISD	−2.5	−2.9	−2.0	39.9	72.3	1.4	−3.1	−3.4	−2.8	218.7	257.5	11.2	−3.3	−3.7	−2.9	372.6	483.4	24.2
200	NRP-IS	−2.2	−2.5	−1.9	35.7	56.5	0.6	−2.6	−3.0	−2.3	203.0	198.4	7.5	−2.7	−3.0	−2.4	360.7	367.8	16.9
	NRP-ISD	−11.8	−12.3	−11.2	44.1	74.8	1.3	−12.6	−13.3	−11.5	206.3	254.8	11.1	−13.0	−13.7	−11.7	391.7	480.5	24.1
250	NRP-IS	−4.0	−4.4	−3.3	35.2	51.1	0.4	−4.5	−4.9	−4.1	214.6	179.0	5.8	−4.7	−5.0	−4.4	426.7	325.0	13.7
	NRP-ISD	−12.6	−13.2	−11.9	45.6	49.9	0.5	−13.6	−14.1	−12.9	229.4	184.4	6.9	−13.9	−14.2	−13.4	422.8	344.7	16.2
300	NRP-IS	−7.6	−8.0	−7.2	36.3	52.4	0.5	−8.0	−8.4	−7.6	207.0	188.2	6.2	−8.2	−8.4	−7.7	422.6	350.2	14.5
	NRP-ISD	−26.9	−27.3	−26.3	39.9	51.0	0.3	−27.8	−28.7	−26.7	233.8	181.0	6.6	−28.1	−28.8	−27.1	497.5	337.3	15.4
350	NRP-IS	−9.1	−9.5	−8.7	35.5	47.1	0.2	−9.6	−10.1	−9.3	231.7	171.2	5.1	−9.8	−10.3	−9.4	488.0	311.6	12.5
	NRP-ISD	–	–	–	46.9	50.3	0.2	–	–	–	256.5	178.2	6.1	–	–	–	510.2	328.9	14.8

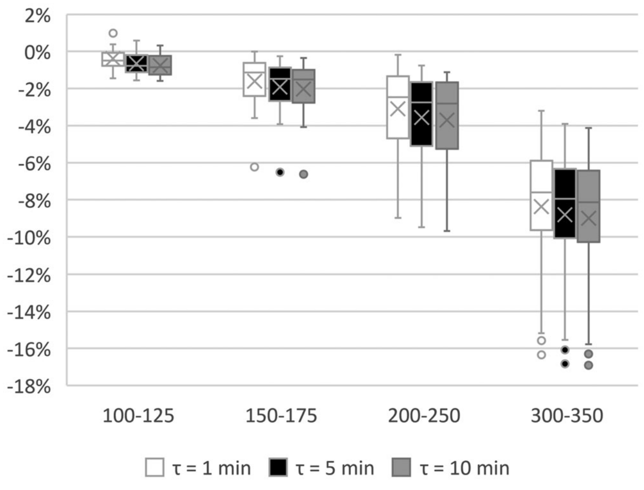

Fig. 1. Δ_{obj}^{avg} for the NRP-IS.

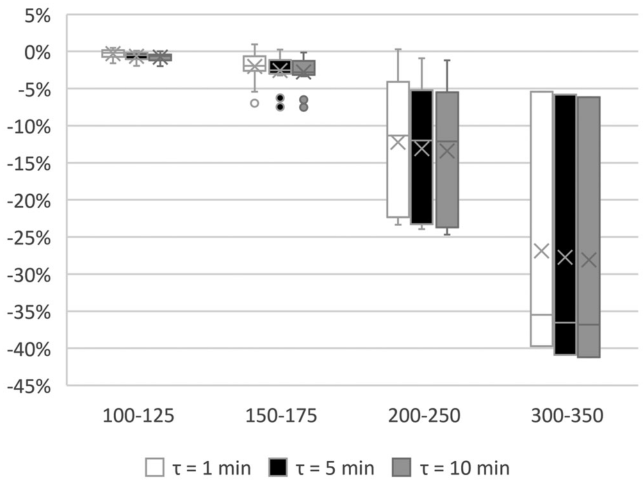

Fig. 2. Δ_{obj}^{avg} for the NRP-ISD.

solution is at least 70% of the total, both for NRP-IS and NRP-ISD. Within 1 minute, the percentage is about 70–75% for instances with 100,125 requests. The average over the two variants rises with the increase in the number of requests, taking values of about 73%, 93%, 98%, and eventually 100%. For 5 and 10 minutes, results are never below 90%. Generally, `ALNS-KS` is always able to outperform `EXACT` in instances with more than 200 requests.

Again from Table 6, we can observe how in both variants, the *ttb* is in general above half of the available time. This is a sign that `ALNS-KS` is able to be more efficient as τ increases. It is also evident that, regardless of τ, more time is needed to find the best solution for the NRP-ISD. This could be due to the fact that the minimum demand constraints lead the `ALNS-KS` to have a higher number of iterations in which the solution repaired is discarded since not feasible. Interesting enough, KS

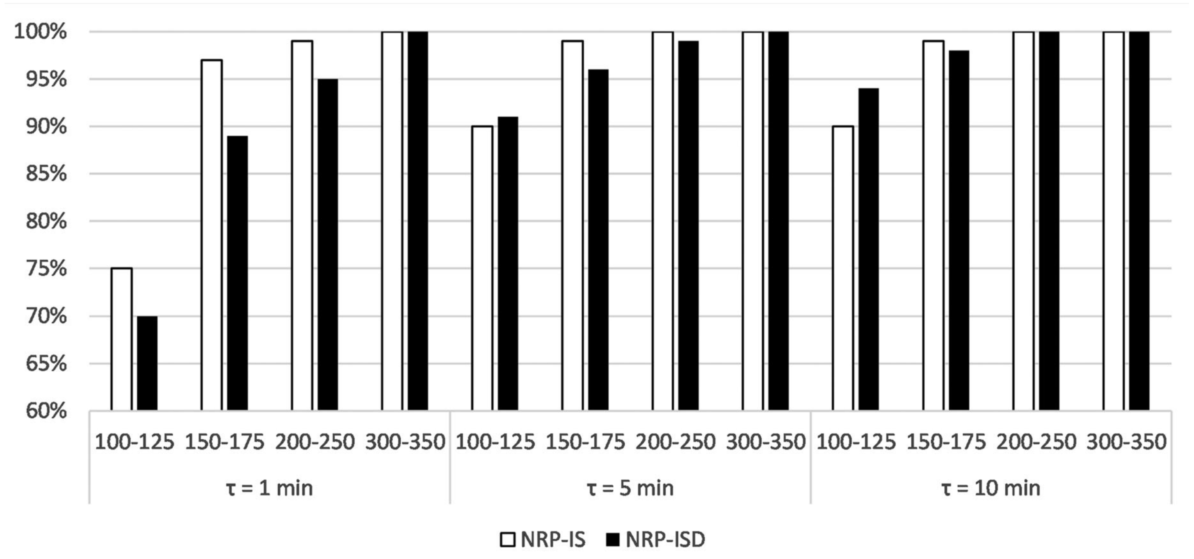

Fig. 3. Percentage of `ALNS-KS` runs that outperform `EXACT`.

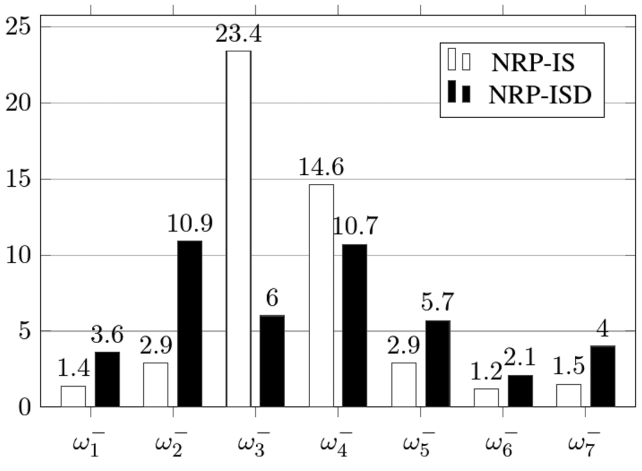

Fig. 4. Average number of s_B improvements by the destroy operators.

is called in proportion much more for $\tau = 10$. This is due to the fact that ALNS is able to find improving solutions at the beginning of the execution while it more likely stalls afterwards.

Finally, in Figs. 4 and 5, we show the contribution given by the destroy and repair operators to `ALNS-KS` in 10 minutes. In particular, we show the average number of times in which, in a run, the specific operator led to an improvement of the best incumbent solution s_B. For NRP-IS, the most successful destroy operator is ω_3^-, with an average number of improvements of s_B per run of 23.4. Next, we find ω_4^-, with 14.6 average improvements. The other destroy operators stand at values below 3, so they probably do not contribute significantly to the achievement of the final solution.

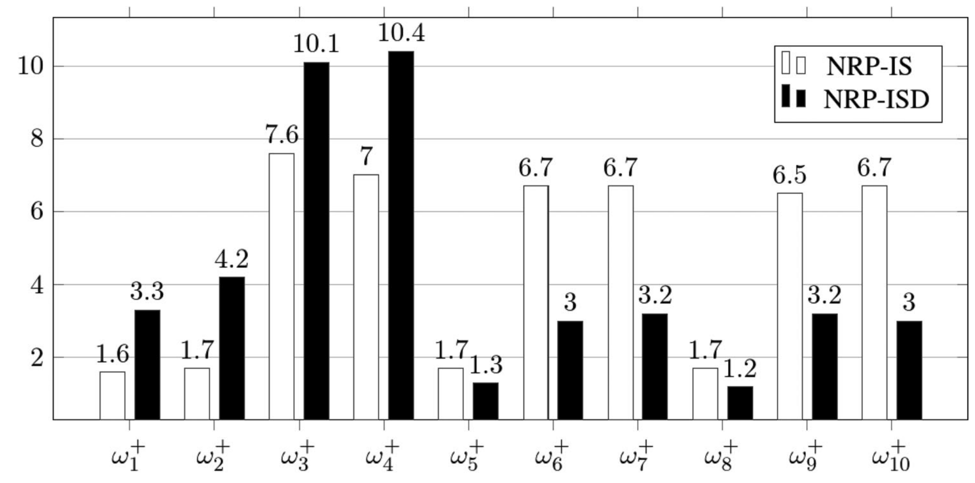

Fig. 5. Average number of s_B improvements by the repair operators.

Nevertheless, as some of these operators are designed as a way to achieve diversification (ω_6^- and ω_7^- in particular), we believe they still play a crucial role in our implementation. For NRP-ISD, the most efficient destroy operators are ω_2^- and ω_4^-, with an average number of improvements of about 11. Next, we find ω_5^- with 5.7. Unlike the NRP-IS, ω_3^- (which sorts the requests by profit over service time) does not appear to be particularly effective for the variant with minimum demand. In general, no destroy operator stands out among the others probably because the implemented sorting procedures do not consider the minimum demand constraints, thus generating non-feasible solutions. Concerning repair operators, six operators (ω_3^+, ω_4^+, ω_6^+, ω_7^+, ω_9^+, ω_{10}^+) obtain for NRP-IS a very similar performance of about seven average improvements of s_B, while the remaining do not have a significant impact. For the NRP-ISD variant, instead, ω_3^+ and ω_4^+ stand out with about 10 improvements of s_B for each run, relegating all the other repair operators to a marginal role. Interesting enough, ω_1^- and ω_1^+, that is, the random operators, do not particularly affect either of the two variants, thus justifying the effort for deepening the specific problem features.

6.3.3. Evaluation of the KS performance

In order to evaluate the effectiveness of KS within our framework, we perform two different analyses. First, we evaluate the impact of KS during the execution in terms of best solution improvements, and then we compare the results of `ALNS-KS` with the results obtained by a pure ALNS.

In Fig. 6, we report the average number of improvements of the current best solution produced by KS for the NRP-IS and NRP-ISD variants. We consider both improvements in terms of total duration of working time for the nurses, without any change in objective function value (# *route improv.*) and improvements of total profit collected (# *obj improv.*). The Figure reports values for τ =1, 5, and 10 minutes, aggregated by number of requests. A clear upward trend emerges from the two charts, which confirms the fact that KS is more effective for hard-to-solve instances. In the NRP-IS variant, the average number of objective function improvements within 10 minutes goes from 2.3 for 100–125 requests to 3.5 for 300–350 ones. There is no clear trend for the number

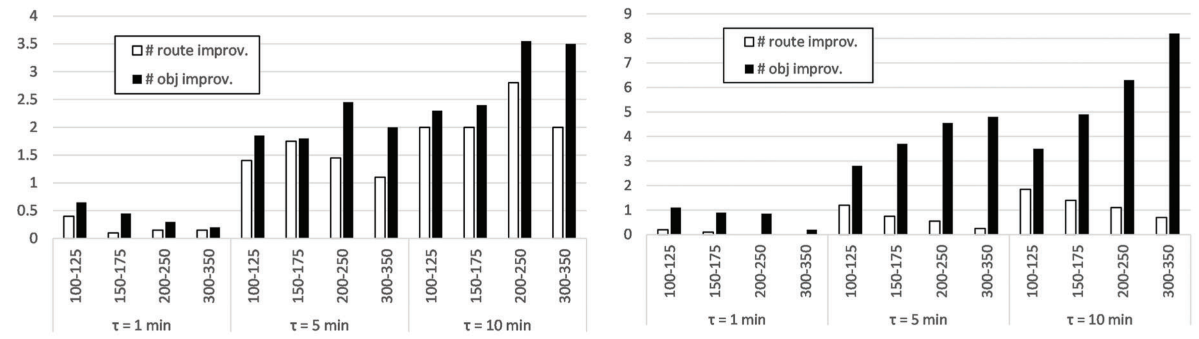

Fig. 6. Average number of s_B improvements in terms of routing cost and total profit produced by KS for NRP-IS (left) and NRP-ISD (right).

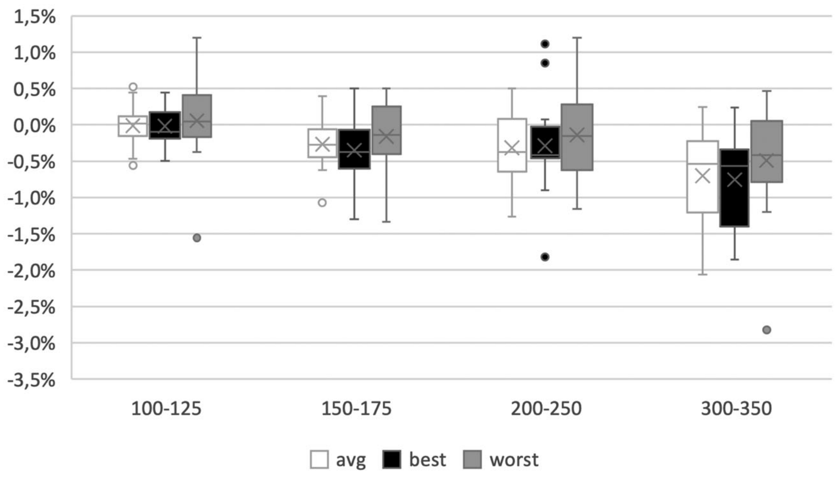

Fig. 7. `ALNS-KS` versus `ALNS-noKS` for NRP-IS. Average, best, and worst objective function gap.

of route improvements, which stays between 2 and 3. The impact of KS is even more clear when considering NRP-ISD. In this case, the average number of best objective improvement goes from 3.5 (100–150 requests) to 8.2 (300–350). This is because a matheuristic can contribute more to the overall effectiveness of an hybrid method when it is more difficult to identify feasible solutions. Note that, the slight downward trend shown for one minute is because `ALNS-KS` can easily find improving solutions early in the algorithm execution and a larger number of requests implies a lower number of iterations, thus reducing the number of KS calls.

In Figs. 7–9, we present the comparison between `ALNS-KS` and its variant not including the KS intensification (`ALNS-noKS` hereafter). In Fig. 7, we show the boxplots over all the NRP-IS instances of the gap between the objective function value obtained by the two algorithms in the average, best, and worst case over five runs. Note that, in order to perform a fair comparison, we kept the same settings for `ALNS-KS` both in the NRP-IS and NRP-ISD cases, while a precise tuning of $thrEpImp$ in particular would have probably helped achieving better results for the NRP-IS variant. The results show no clear trends and, even if the average gap calculated considering all instances is slightly in

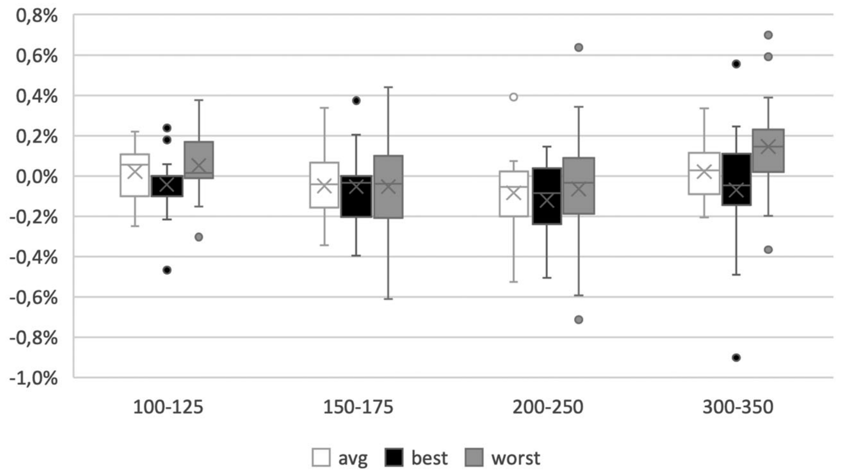

Fig. 8. `ALNS-KS` versus `ALNS-noKS` for NRP-ISD. Average, best, and worst objective function gap.

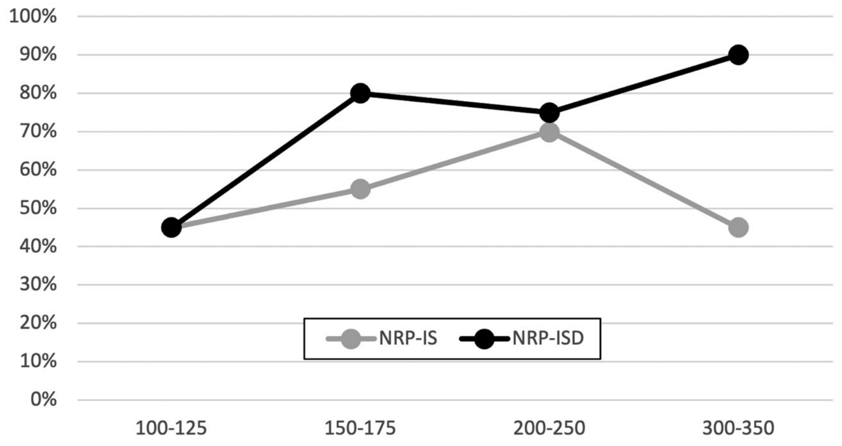

Fig. 9. Percentage of instances where `ALNS-KS` is better than `ALNS-noKS`.

favor of `ALNS-KS` (-0.02%), for certain values of $|N|$, `ALNS-noKS` performs better sometimes (e.g., the gap is 0.02% for 300–350 requests). Although the best value obtained is almost always better for `ALNS-KS`, `ALNS-noKS` is often able to obtain a better performance in the worst case. This implies that, in instances in which finding feasible solutions is quite easy (like the ones without demand), `ALNS-noKS` can be good enough. In Fig. 8, we report the comparison of the same methods for the NRP-ISD. In this case, we can clearly see that KS is much more effective (-0.3% on average), and a downward trend emerges, indicating how, especially in the 300–350 requests instances, the hybridization has its advantages. For these instances, `ALNS-KS` obtains, on average, solutions that are 0.7% better than the `ALNS-noKS` ones, which corresponds to an improvement of about 100 units of profit. In most cases, `ALNS-noKS` struggles to quickly find high-quality feasible solutions, thus, unlike in the NRP-IS case, substituting thousands of ALNS iterations for a few KS runs consistently produces improvements. Finally, in Fig. 9, we report the percentage of instances in which the average objective function value over five runs obtained by `ALNS-KS` is better than the `ALNS-noKS` one. These results confirm that KS is more effective for the NRP-ISD variant. If we consider 300–350 requests, `ALNS-KS` is better than `ALNS-noKS` in 90% of the cases. However, even for NRP-IS, for 200–250 requests, the percentage of `ALNS-KS` improved instances is 70%.

7. Managerial insights

In this Section, we conduct a sensitivity analysis by exploiting the ability of our solution approach to solve realistic-size HHC working scenarios. In particular, we investigate, through several indicators, which is the impact produced by increasing the number of incompatibilities and of available nurses while keeping the same set of requests.

To this aim, we consider 10 instances (five for each variant of the problem) involving 4 nurses, 50 patients, 5 service types, 100 requests, and a single pair of incompatible service types. Then, for each basic instance, we

1. increase the number $|F|$ of nurses from 4, to 5, and to 6;
2. increase the number $|B|$ of pairs of incompatible service types from 1, to 2, and to 3.

Each instance is solved to optimality by first running our `ALNS-KS` for 10 minutes and then by running our branch-and-cut `EXACT` initialized with the best solution found by the heuristic (to be precise, we stop the computation when the potential error between the value of the objective function and the real optimum of the problem is certified to be less than 1%). The assessment is done by calculating, for each instance and for each change, the following KPIs:

- Δobj: the percentage of improvement in terms of objective function value with respect to the basic instance;
- $tfree$: the percentage of working time for the employed nurses that remains unused with respect to the total working time available;
- *rsat*: the percentage of fulfilled requests with respect to the total number of requests;
- *rsat/p*: the average percentage of fulfilled requests with respect to the total ones coming from a single patient;
- *psat*: the percentage of patients whose requests have been totally satisfied.

Figures 10 and 11 show the values of the above KPIs (averaged on all the tested instances) with respect to the increment of $|F|$ and of $|B|$, respectively. Given the magnitude of the values, the left charts report on Δobj and $tfree$, whereas the remaining KPIs are shown in the right charts. Moreover, for each KPI, the dotted line represents the value concerning NRP-IS instances, whereas the straight line represents the value concerning the NRP-ISD.

Considering Fig. 10, it is clear that all the KPIs increase steadily and almost linearly as the number of available nurses increases. Moreover, there is no sensible difference in the increasing between the NRP-IS case and the NRP-ISD one. From the left chart, it appears that any new available nurse allows to gain about 8–10% in terms of total profit collected, while the percentage of total free time slightly increases by about 1%. This is reasonable, since the objective function of the problem only aims at maximizing the profit without caring about saving time. However, even

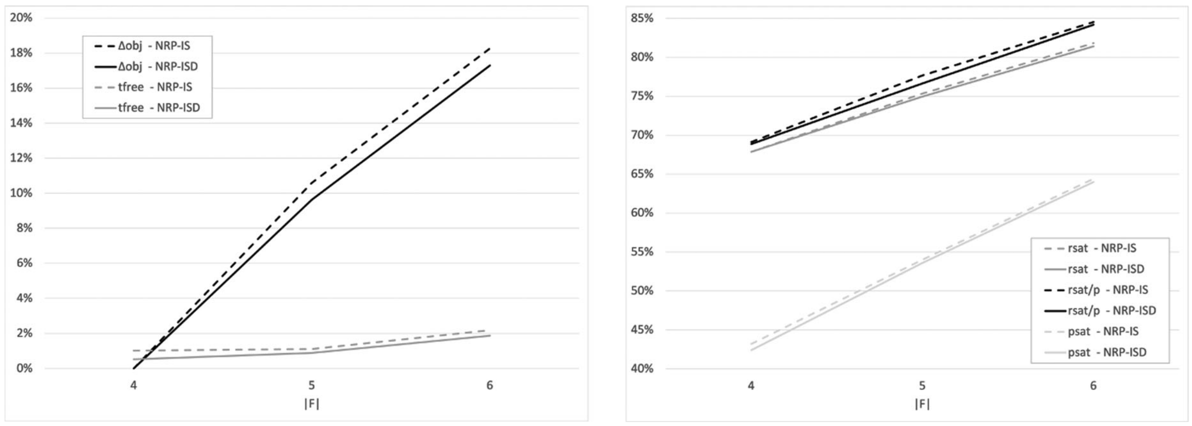

Fig. 10. KPIs trends for increasing values of $|F|$.

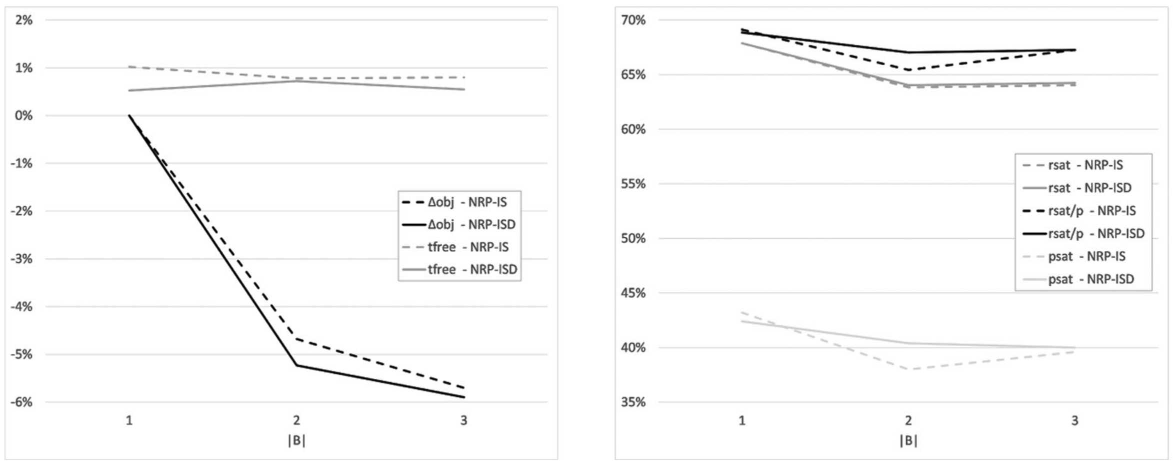

Fig. 11. KPIs trends for increasing values of $|B|$.

if in small extent, a little more time is much appreciated as it allows for more flexibility and to better deal with unexpected events. The right chart, instead, shows that the introduction of a new nurse increases the percentage of fulfilled requests (both overall and per patient) by 7–8% and the percentage of totally satisfied patients by about 10%. Clearly, this analysis is useful to evaluate, at a tactical level, the dimension of the fleet of nurses depending on the additional cost of the personnel, the magnitude of the profit, and the quality-of-service desired.

Figure 11 shows that, in general and as expected, the increase of service type incompatibilities negatively affects the KPIs considered. This is clear from the left chart, where we see that introducing an additional incompatibility drops down the total collected profit of about 5% (the drop is a bit more consistent in the NRP-ISD case), while the introduction of a further incompatibility just reduces the profit of 1%. However, this negative trend has some exceptions. First, the $tfree$

KPI remains almost stable with little up-and-down oscillations of at most 0.5% (remember that the time minimization is not directly pursued by the problem). More interesting, and in particular for the case without minimum demand, the $rsat/p$ and the $psat$ KPIs increase of about 2–3% when increasing the number of incompatibilities from 2 to 3. Although counterintuitive, this can be explained by the fact that the additional incompatibility may affect some patients that are not anymore convenient to visit in terms of profit, thus allowing to better concentrate on other patients. The just carried out analysis, and in particular the signal on the possible profit losses, may help the management of the nurse companies in deciding which services to consider in their basket.

8. Conclusions

In this paper, we study a Nurse Routing Problem in which a set of nurses needs to be scheduled and routed to perform services to patients with the aim of maximizing the collected profit. This setting is able to capture more realistic features than the classical VRP extensions studied in the specialized literature. Moreover, our problem is further characterized by different service types, working time limitations for the nurses, service incompatibilities, and minimum demand. We first provide a new compact MILP formulation, eventually strengthened by the introduction of different valid inequalities. Then, a hybrid ALNS is developed to address the most complex working scenarios. Apart from the specific destroy and repair operators, our ALNS approach has been enriched by a Kernel Search-based intensification procedure exploiting the exact solution of MILP restricted problems. An extensive campaign of computational experiments has been conducted to validate and assess both the new model and the proposed `ALNS-KS` approach. In particular, we test two variants of the problem, with and without a minimum demand to be guaranteed for each type of service. In both cases, our new formulation outperforms the existing one, thus providing new benchmarks. However, both exact models are not suitable against medium and large instances, whereas our `ALNS-KS` has proved to be able to provide high-quality solutions in very short time even for the hardest scenarios and to consistently outperform the exact formulations.

Some future research directions can be foreseen. In particular, given the efficiency achieved by the proposed matheuristic, it could be interesting to extend the problem to a multi-period setting, thus planning day-by-day operations over a longer horizon (e.g., a week) and embedding medium-term economies of scale. In such a setting, it is reasonable that the requests of a patient not satisfied during a day are likely to receive a higher priority (profit) in the next days. Moreover, to obtain robust home healthcare services, also uncertainties and possible disruptions should be explicitly taken into account. In this case, apart from the classical issues linked to traffic in road networks, also delays on the service times (e.g., due to unexpected reactions or difficulties during the treatment) become important.

Acknowledgements

Open Access Funding provided by Università degli Studi di Brescia within the CRUI-CARE Agreement.

References

Akjiratikarl, C., Yenradee, P., Drake, P.R., 2007. PSO-based algorithm for home care worker scheduling in the UK. *Computers & Industrial Engineering* 53, 4, 559–583.

Allaoua, H., Borne, S., Létocart, L., Wolfler Calvo, R., 2013. A matheuristic approach for solving a home health care problem. *Electronic Notes in Discrete Mathematics* 41, 471–478.

Angelelli, E., Mansini, R., Speranza, M.G., 2010. Kernel search: a general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research* 37, 11, 2017–2026.

Angelelli, E., Mansini, R., Speranza, M.G., 2012. Kernel search: a new heuristic framework for portfolio selection. *Computational Optimization and Applications* 51, 1, 345–361.

Assunção, L., Mateus, G.R., 2021. Coupling feasibility pump and large neighborhood search to solve the Steiner team orienteering problem. *Computers & Operations Research* 128, 105175.

Bachouch, R.B., Guinet, A., Hajri-Gabouj, S., 2011. A decision-making tool for home health care nurses' planning. *Supply Chain Forum: An International Journal* 12, 1, 14–20.

Beraldi, P., Bruni, M.E., Manerba, D., Mansini, R., 2017. A stochastic programming approach for the traveling purchaser problem. *IMA Journal of Management Mathematics* 28, 1, 41–63.

Bertels, S., Fahle, T., 2006. A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers & Operations Research* 33, 10, 2866–2890.

Bettinelli, A., Cacchiani, V., Malaguti, E., 2017. A branch-and-bound algorithm for the knapsack problem with conflict graph. *INFORMS Journal on Computing* 29, 3, 457–473.

Bianchessi, N., Mansini, R., Speranza, M.G., 2018. A branch-and-cut algorithm for the team orienteering problem. *International Transactions in Operational Research* 25, 2, 627–635.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 9, 1124–1137.

Braekers, K., Hartl, R.F., Parragh, S.N., Tricoire, F., 2016. A bi-objective home care scheduling problem: analyzing the trade-off between costs and client inconvenience. *European Journal of Operational Research* 248, 2, 428–443.

Bräysy, O., Dullaert, W., Nakari, P., 2009. The potential of optimization in communal routing problems: case studies from Finland. *Journal of Transport Geography* 17, 6, 484–490.

Bredstrom, D., Ronnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research* 191, 1, 19–31.

Chahed, S., Marcon, E., Sahin, E., Feillet, D., Dallery, Y., 2009. Exploring new operational research opportunities within the home care context: the chemotherapy at home. *Health Care Management Science* 12, 2, 179–191.

Cinar, A., Salman, F.S., Bozkaya, B., 2021. Prioritized single nurse routing and scheduling for home healthcare services. *European Journal of Operational Research* 289, 3, 867–878.

Colombi, M., Corberán, Á., Mansini, R., Plana, I., Sanchis, J.M., 2017. The directed profitable rural postman problem with incompatibility constraints. *European Journal of Operational Research* 261, 2, 549–562.

Department of Economic and Social Affairs of the United Nations Secretariat, 2017. World Population Prospects: The 2017 Revision. https://esa.un.org/unpd/wpp/Publications/Files/WPP2017_KeyFindings.pdf.

Di Mascolo, M., Espinouse, M., Ait Haddadene, S., 2018. Taking patients wishes into account for daily planning in the home health care contex. *IFAC-PapersOnLine* 51, 11, 1010–1015.

Di Mascolo, M., Espinouse, M.L., El Hajri, Z., 2017. Planning in home health care structures: a literature review. *IFAC-PapersOnLine* 50, 1, 4654–4659.

Dohn, A., Kolind, E., Clausen, J., 2009. The manpower allocation problem with time windows and job-teaming constraints: a branch-and-price approach. *Computers & Operations Research* 36, 4, 1145–1157.

Eveborn, P., Flisberg, P., Ronnqvist, M., 2006. Laps care: an operational system for staff planning of home care. *European Journal of Operational Research* 171, 3, 962–976.

Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R., 2018. A bi-objective green home health care routing problem. *Journal of Cleaner Production* 200, 423–443.

Fikar, C., Hirsch, P., 2015. A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production* 105, 300–310.

Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: a review. *Computers & Operations Research* 77, 86–95.

Gendreau, M., Manerba, D., Mansini, R., 2016. The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: a branch-and-price approach. *European Journal of Operational Research* 248, 1, 50–71.

Gobbi, A., Manerba, D., Mansini, R., Zanotti, R., 2019. A kernel search for a patient satisfaction-oriented nurse routing problem with time-windows. *IFAC-PapersOnLine* 52, 13, 1669–1674.

Goossens, D., Spieksma, F.C., 2009. The transportation problem with exclusionary side constraints. *4OR* 7, 1, 51–60.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research* 84, 116–126.

Hanafi, S., Mansini, R., Zanotti, R., 2020. The multi-visit team orienteering problem with precedence constraints. *European Journal of Operational Research* 282, 2, 515–529.

Hiermann, G., Prandtstetter, M., Rendl, A., Puchinger, J., Raidl, G.R., 2015. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research* 23, 1, 89–113.

Hindle, T., Hindle, A., Spollen, M., 2000. Resource allocation modelling for home-based health and social care services in areas having differential population density levels: a case study in Northern Ireland. *Health Services Management Research* 13, 3, 164–169.

Hindle, T., Hindle, G., Spollen, M., 2009. Travel-related costs of population dispersion in the provision of domiciliary care to the elderly: a case study in English Local Authorities. *Health Services Management Research* 22, 1, 27–32.

Koeleman, P.M., Bhulai, S., Van Meersbergen, M., 2012. Optimal patient and personnel scheduling policies for care-at-home service facilities. *European Journal of Operational Research* 219, 3, 557–563.

Lamanna, L., Mansini, R., Zanotti, R., 2022. A two-phase kernel search variant for the multidimensional multiple-choice knapsack problem. *European Journal of Operational Research* 297, 1, 53–65.

Landers, S., Madigan, E., Leff, B., Rosati, R.J., McCann, B.A., Hornbake, R., MacMillan, R., Jones, K., Bowles, K., Dowding, D., Lee, T., Moorhead, T., Rodriguez, S., Breese, E., 2016. The future of home health care: a strategic framework for optimizing value. *Home Health Care Management and Practice* 28, 4, 262–278.

Lasfargeas, S., Gagné, C., Sioud, A., 2019. Solving the home health care problem with temporal precedence and synchronization. In Talbi, E., Nakib, A. (eds) *Bioinspired Heuristics for Optimization*, *Studies in Computational Intelligence*, Vol. 774. Springer, Berlin, pp. 251–267.

Maffioli, F., Schiomachen, A., 1997. A mixed-integer model for solving ordering problems with side constraints. *Annals of Operations Research* 69, 277–297.

Manerba, D., Mansini, R., 2015. A branch-and-cut algorithm for the multi-vehicle traveling purchaser problem with pairwise incompatibility constraints. *Networks* 65, 2, 139–154.

Manerba, D., Mansini, R., 2016. The nurse routing problem with workload constraints and incompatible services. *IFAC-PapersOnLine* 49, 12, 1192–1197. 8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM) Troyes, France, 28–30 June 2016.

Manerba, D., Mansini, R., Riera-Ledesma, J., 2017. The traveling purchaser problem and its variants. *European Journal of Operational Research* 259, 1, 1–18.

Mankowska, D.S., Meisel, F., Bierwirth, C., 2014. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science* 17, 15–30.

Mansini, R., Zanotti, R., 2020. Optimizing the physician scheduling problem in a large hospital ward. *Journal of Scheduling* 23, 3, 337–361.

Martinez, C., Espinouse, M., Di Mascolo, M., 2018. Continuity of care in home services: a client-centered heuristic for the home health care routing and scheduling problem. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1045–1050.

Mutingi, M., Mbohwa, C., 2014. Multi-objective homecare worker scheduling: a fuzzy simulated evolution algorithm approach. *IIE Transactions on Healthcare Systems Engineering* 4, 4, 209–216.

Nasir, J.A., Dang, C., 2018. Solving a more flexible home health care scheduling and routing problem with joint patient and nursing staff selection. *Sustainability (Switzerland)* 10, 1, 148.

Nolan, M., 1999. 'It's not the same as him being at home': creating caring partnerships following nursing home placement. *Journal of Clinical Nursing* 8, 6, 723–730.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* 34, 8, 2403–2435.

Rasmussen, M.S., Justesen, T., Dohn, A., Larsen, J., 2012. The home care crew scheduling problem: preference-based visit clustering and temporal dependencies. *European Journal of Operational Research* 219, 3, 598–610.

Redjem, R., Marcon, E., 2016. Operations management in the home care services: a heuristic for the caregivers' routing problem. *Flexible Services and Manufacturing Journal* 28, 280–303.

Rest, K.D., Trautsamwieser, A., Hirsch, P., 2012. Trends and risks in home health care. *Journal of Humanitarian Logistics and Supply Chain Management* 2, 1, 34–53.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40, 4, 455–472.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* 159, 2, 139–171.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In Maher, M., Puget, J.F. (eds) *Principles and Practice of Constraint Programming — CP98*, *Lecture Notes in Computer Science*, Vol. 1520. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 417–431.

Şuvak, Z., Altınel, İ.K., Aras, N., 2020. Exact solution algorithms for the maximum flow problem with additional conflict constraints. *European Journal of Operational Research* 287, 2, 410–437.

Trautsamwieser, A., Gronalt, M., Hirsch, P., 2011. Securing home health care in times of natural disasters. *OR Spectrum* 33, 3, 787–813.

Trautsamwieser, A., Hirsch, P., 2011. Optimization of daily scheduling for home health care services. *Journal of Applied Operational Research* 3, 3, 124–136.

Vansteenwegen, P., Gunawan, A., 2019. Definitions and mathematical models of single vehicle routing problems with profits. In Vansteenwegen, P., Gunawan, A. (eds) *Orienteering Problems: Models and Algorithms for Vehicle Routing Problems with Profits*. Springer International Publishing, Cham, pp. 7–19.

Yalçindağ, S., Matta, A., Şahin, E., Shanthikumar, J.G., 2016. The patient assignment problem in home health care: using a data-driven method to estimate the travel times of care givers. *Flexible Services and Manufacturing Journal* 28, 304–335.