Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-13 (2019) 1669–1674

A Kernel Search for a Patient Satisfaction-oriented Nurse Routing Problem with Time-Windows

A. Gobbi * **D. Manerba** ** **R. Mansini** * **R. Zanotti** *

* *Department of Information Engineering, University of Brescia, via Branze 38, 25123, Brescia, Italy (e-mail: a.gobbi001@unibs.it, renata.mansini@unibs.it, roberto.zanotti@unibs.it)*

** *Department of Control and Computer Engineering and ICT for City logistics and Enterprises (ICE) Lab, Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino, Italy (e-mail: daniele.manerba@polito.it)*

Abstract: We study a variant of the Nurse Routing Problem where each patient may require more than one service at possible different times during the day. Each executed service yields a profit, and an additional reward is gained if all services associated with a patient are fulfilled. The problem looks for nurse routes, each one not exceeding a predefined working time limit, that maximize the global collected service profits plus the patient rewards, while respecting the time windows associated with services. We first provide a compact Mixed-Integer Linear Programming formulation for the problem. Then, we develop an Iterative Kernel Search to solve the problem heuristically. Finally, we compare the heuristic performance on several instances with that of the plain model solved through a state-of-the-art exact solver and strengthened by the separation of valid inequalities. The obtained results clearly show that our heuristic algorithm finds fairly good solutions in terms of quality, despite the use of shorter computational times.

Keywords: Home Health-care, Nurse Routing Problem, Patient satisfaction, Time Windows, Kernel Search.

1. INTRODUCTION

In the so-called Nurse Routing Problem (NRP), a set of nurses are available to visit a set of patients (spread over a geographical area) to perform care services. Each nurse has a predefined working time limit, which cannot be exceeded considering both the time required to serve the patients and that to travel among them. In this paper, we introduce and study a new NRP variant in which the patients require different services during possible different time intervals along the day, i.e., multiple time windows are considered. The problem objective is to maximize the score (*service profit*) collected for executed services plus additional rewards (*patient profits*), obtained only if all the services associated with a given patient are completed. We call this problem the Patient Satisfaction-oriented Nurse Routing Problem with Time-Windows (PSNRP-TW).

The rest of this paper is organized as follows. In Section 2, we present an overview of the most recent papers in the area of home health care, highlighting those related to ours. In Section 3, we formally describe and formulate the PSNRP-TW while, in Section 4, we present the Iterative Kernel Search heuristic framework adopted. In Section 5, the generation of instances and the relative computational experiments are discussed. Finally, Section 6 presents conclusions and future research.

2. LITERATURE REVIEW AND RELATED PROBLEMS

Long-term home health care services have experienced and still face a massive demand, steadily and rapidly growing in the last two decades. Nowadays, home care includes a large set of different services, e.g. simple assistance (dressing, bathing, using the bathroom) to the patients, a long list of specific medical operations (dosing drugs, performing injections, or taking vital values), and even very complex tasks such as chemotherapy (Chahed et al., 2009). The importance of providing such services with adequate efficiency and timing, despite of the limited resources available, has motivated the attention of researchers and practitioners in the recent past (Milburn, 2012). Besides several studies about benchmarking nursing performance (Ozcan, 2008), a very large number of specific optimization problems arising in the field has been approached through modeling and algorithmic frameworks. Our paper, studying a variant of the Nurse Routing Problem, belongs to this last stream of research.

A complete literature review on the topic would be out of the scope of this paper, therefore we refer the reader to a very recent and comprehensive survey by Fikar and Hirsch (2017) about nurse routing and scheduling problems. The authors classify the existing works by looking at the con-

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.11.440

straints involved and at the time horizon (single or multi-period) considered in the problems, and by the adopted solution method. However, other interesting papers have appeared more recently, with the general aim of tackling more and more realistic problems through the addition of constraints or optimization objectives. For example, Lin et al. (2018) and Nasir and Dang (2018) have considered skill requirements for the nurses to perform services and time constraints such as time-windows for the visits and working time regulations for the nurses. Fathollahi-Fard et al. (2018) also considered a restricted capacity for the nurses to carry drugs and a bi-objective problem including the minimization of the transportation costs and of the CO_2 emission. Yang et al. (2018) studied a stochastic home health-care routing problem in the case of dense communities, where the distances are typically short while the waiting times are high, and multiple appointments. Finally, Lasfargeas et al. (2019) tackled a very complex setting also including time-windows, nurse/patients exclusions, precedence and synchronization constraints. In these works, the problems are optimized through the use of heuristic and meta-heuristic algorithms (e.g., Variable Neighborhood Search, Harmony Search, Ant Colony Optimization).

The most part of the appeared problems have been studied as variant of the well-known Vehicle Routing Problem (VRP) in which, in general, all the patients need to be visited and the main objective is to minimize the visiting/traveling costs. Instead, in our work, we focus on the quality of the care process by maximizing the profit obtainable from performing the services and, therefore, by possibly sub-selecting which patients to visit. This feature, and the presence of different services to perform, make our problem a variant of the so-called Traveling Purchaser Problem (Manerba et al., 2017, Beraldi et al., 2017), or better, of its multi-vehicle variant (Manerba and Mansini, 2015, Gendreau et al., 2016). The most similar work has been published by Manerba and Mansini (2016), where the authors study a Nurse Routing Problem with workload constraints and incompatibility among services to the same patient. The two exact methods proposed, based on branch-and-cut and branch-and-price, respectively, have been able to optimally solve small and medium-size instances. As in our case, that paper included the patient selection feature, however no time-windows were considered and a less sophisticated scoring function was used for the maximization of the profits.

3. MATHEMATICAL FORMULATION

We propose a compact Mixed-Integer Linear Programming (MILP) formulation for the PSNRP-TW inspired by the model proposed by Maffioli and Sciomachen (1997) in a scheduling context, and then reused for Team Orienteering Problem by Bianchessi et al. (2018) and Hanafi et al. (2018) to exclude subtours. To exploit such a formulation, each node of the graph has to be visited at most once.

Let $M = \{1, ..., m\}$ be the set of patients, F the set of nurses available to serve patients, K the set of service types, and N the set collecting each possible service required by each possible patient. Each service $i \in N$ is associated with exactly one service type $k \in K$, so services required by different patients might be of the same type k .The problem can be defined over a direct graph $G = (V, A)$, with node set $V = N \cup \{0, n+1\}$ and arc set $A = \{(i, j) : i, j \in V, i \neq n+1, j \neq 0, i \neq j\}$. Nodes 0 and $n+1$ are initial and terminal depot, respectively. We assume that set N is partitioned into m disjoint and non-empty subsets $N_h, h = 1, ..., m$, each one associated with a different patient, i.e. $N = \bigcup_{h=1}^{m} N_h$ and $N_h \cap N_{h'} = \emptyset$ for $h \neq h'$. We assume that all the available nurses are sufficiently skilled to perform all the types of services $i \in N$. Each nurse has a maximum daily working time equal to T_{max}. Each service $i \in N$ of same tipe $k \in K$ has a predefined execution time s_i, and a hard time window $[e_i, l_i]$ that represents the interval of time within which service i has to begin. Nodes belonging to the same set N_h are services requested by the same patient, and thus have the same location, but may have different time windows for their execution. A positive profit p_i is associated with each service $i \in N$, whereas a patient reward α_h is assigned to each subset N_h. Such a reward can be collected only if all nodes in N_h have been served. A patient may be visited by more than one nurse, whereas each service can be executed by exactly one nurse. The profit associated with each service can be seen as a positive scalar quantity measuring the importance/priority to fulfill such a service. A similar interpretation can be given to the reward associated to each patient. We define as t_{ij} the nonnegative traveling time from node i to node j, where $t_{ij} = 0$ for all $i, j \in N_h, h = 1, ..., m$. Travel times satisfy the triangle inequality. Each nurse leaves the depot 0 at time 0, and has to terminate his work at node $n+1$ before time T_{max}. Each nurse executes a service as soon as he reaches the associated node (no waiting times are allowed).

We introduce the following sets of binary variables

$$w_i = \begin{cases} 1 \text{ if node } i \in N \text{ is visited;} \\ 0 \text{ otherwise.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 \text{ if arc } (i,j) \text{ is traversed;} \\ 0 \text{ otherwise.} \end{cases}$$

$$y_h = \begin{cases} 1 \text{ if all nodes in } N_h \text{ are visited;} \\ 0 \text{ otherwise.} \end{cases}$$

We also define a set of continuous variables $z_{ij}, (i, j) \in A \setminus \{(0, n+1)\}$ indicating the arrival time in node j when arriving from i. Finally, given a set $S \subset V$, let $\delta^+(S)$ be the set of arcs (i, j) with $i \in S$ and $j \in V \setminus S$ and $\delta^-(S)$ be the set of arcs (i, j) with $j \in S$ and $i \in V \setminus S$.

The PSNRP-TW can be modeled as follows:

$$\max \quad \sum_{i \in N} p_i w_i + \sum_{h \in M} \alpha_h y_h \tag{1}$$

subject to

$$\sum_{(i,j) \in \delta^+(\{i\})} x_{ij} = \sum_{(j,i) \in \delta^-(\{i\})} x_{ji} = w_i \quad i \in N \tag{2}$$

$$\sum_{(0,j) \in \delta^+(\{0\})} x_{0j} = \sum_{(j,n+1) \in \delta^-(\{n+1\})} x_{j,n+1} \leq |F| \tag{3}$$

$$\sum_{(i,j)\in\delta^+(\{i\})} z_{ij} - \sum_{(j,i)\in\delta^-(\{i\})} z_{ji} = \sum_{(i,j)\in\delta^+(\{i\})} (t_{ij}+s_i)x_{ij} \quad i \in N \quad (4)$$

$$z_{0j} = t_{0j}x_{0j} \quad (0,j) \in \delta^+(\{0\}) \quad (5)$$

$$(t_{0i}+t_{ij}+s_i)x_{ij} \le z_{ij} \le T_j x_{ij} \quad (i,j) \in A\backslash\{(0,n+1)\} \quad (6)$$

$$e_j x_{ij} \le z_{ij} \le l_j x_{ij} \quad (i,j) \in A \setminus \{(0,n+1)\} \quad (7)$$

$$y_h \le w_i \quad h = 1,...,m, \ i \in N_h \quad (8)$$

$$z_{ij} \ge 0 \qquad (i,j) \in A \setminus \{(0,n+1)\} \quad (9)$$

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

$$w_i \in \{0,1\} \qquad i \in N \quad (11)$$

$$y_h \in \{0,1\} \qquad h = 1,...,m \quad (12)$$

Objective function (1) maximizes the sum of global collected profits earned by performing home care services plus the total patients rewards. Constraints (2) are classical pairing conditions, whereas constraints (3) control the creation of at most $|F|$ routes. Constraints (4) replace the standard subtour elimination constraints and ensure that, if node j is visited immediately after node i, then the time elapsed between the arrival times in the two nodes has to be equal to the service time s_i at node i plus the travel time t_{ij} between i and j. Constraints (5) set a bound on the minimum time required to reach the first node after the depot, whereas constraints (6) define lower and upper bounds on the duration of each route, where parameter T_j is computed as $T_{max} - s_j - t_{j,n+1}$. Constraints (7) impose that service j has to be executed within its time window. Constraints (8) allows variable y_h to be 1 if and only if every service requested by patient h has been performed (no w_i variable, $i \in N_h$, is equal to zero). Constraints (9)–(12) define non-negative condition on z variables and binary conditions on remaining ones.

Note that, in the special case where each patient requires exactly one service, the PSNRP-TW reduces to the well-known Team Orienteering Problem with Time Windows (Vansteenwegen et al., 2009). This shows that PSNRP-TW is $\mathcal{NP}$-hard.

3.1 Valid inequalities

To strengthen the proposed formulation, we add a valid inequality on the global time as follows:

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \le |F|T_{max}.$$

Moreover, even though constraints (4)–(6) ensure that there are no subtours and each route is connected to the depot, we dynamically separate the following Generalized Connectivity Constraints (GCCs):

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \ge w_k, \quad S \subset N, |S| \ge 2, k \in S.$$

Let $(\mathbf{w}^{LP}, \mathbf{x}^{LP}, \mathbf{y}^{LP}, \mathbf{z}^{LP})$ be the optimal solution of the continuous relaxation of problem (1)–(12). A Min-Cut problem is solved at most once for each patient h by selecting the node $l = \arg\max_{i\in N_h}\{w_i^{LP}\}$ with $w_l^{LP} \neq 0$.

4. SOLUTION ALGORITHM

Kernel Search (KS) is a general purpose heuristic framework for the solution of MILP problems, originally developed by Angelelli et al. (2010, 2012). The main idea consist in solving a sequence of MILP subproblems in order to continuously improve the objective function value. A key concept in the construction of subproblems is the one of *item*: an item is an element of the problem for which it has to be decided whether to select it or not in the solution. Restricting the set of items that can be included in the solution helps reducing the solution space, thus decreasing solution time. The *kernel* is the set of promising items (i.e., the items that are more likely to lead to a high quality solution), that is used as a starting point for the construction of all the subproblems. The remaining items are split in several subsets, called *buckets*. The algorithm initially solves a subproblem considering only the items in the kernel, and then proceeds to sequentially solve several subproblems constructed by considering the kernel plus one bucket at a time. After all the buckets have been considered, new iterations of the kernel search might be performed, considering the same set of buckets, or a differently constructed set. In our work, we consider performing a service required by a patient as an item, and thus each item is associated with a variable w_i, $i \in N$. From now on, the term item is used as a synonym of variable associated with such item.

In our implementation, we resort to an Iterative Kernel Search (IKS), a KS variant introduced by Hanafi et al. (2018). Here, items are sorted in non-increasing order of the value they take in the optimal solution of the linear programming relaxation of the problem at hand. Items with the same value are then sorted in non-decreasing order of their reduced cost value. The buckets are iterated twice. In the second iteration, if a patient has only one service that is not performed according to the current best solution, the variable associated to such service is moved in the top bucket. By performing this change, we try to increase the chance to obtain a solution with higher completion rewards.

5. COMPUTATIONAL RESULTS

To test our solution approaches, 40 instances have been generated to simulate several real working scenarios. The locations of the patients and of the depot have been randomly generated into a square with an edge of $30km$ length. Travel times have been computed considering an average speed of $60km/h$. Maximum working time T_{max} is equal to 6 hour. In the first 20 instances, the number of services requested by half of the patients is generated between 1 and 5, while it is generated between 1 and θ for the other half. In all the instances, the number of services requested by half of the patients is generated between 1 and $|K|$, while it is generated between 1 and θ for the other half. Services are classified according to their execution time s_i into short duration (between 5 and 15 minutes), medium duration (between 15 and 45 minutes) and high duration (between 45 and 95 minutes) services. Each duration s_i is randomly generated within one of these three ranges. During the generation process, we make sure that the number of short, medium, and long services does not exceed 60 percent, 40 percent, and 20 percent of the total number of services, respectively. At most fifty percent of the services have a hard time window randomly selected among [0,120], [120,240], and [240,360]. The profit $p_i, i \in N$ is a positive scalar in the range $[2, 200]$. The value 200 characterizes an urgent medical service for a patient. For each patient $h = 1, ..., m$, the reward α_h is set as a percentage γ of the total sum of profits of the services required by the patient himself (i.e., $\alpha_h = \gamma \sum_{i \in N_h} p_i$).

The experiments have been run on a Intel Core i7 5930k machine with 64GB of RAM and running Windows 7 operating system. The proposed mathematical formulation has been solved by using Gurobi, version 8.1.0 [1]. A computational time threshold of one hour has been considered for each instance. The solution used in the following analysis is the best one found by Gurobi within that time limit. With respect to our IKS implementation, we used the top 20% of the sorted list of items as the kernel set, and $\frac{|N|}{5}$ as the bucket size. The global time limit used is 600 seconds, with 480 seconds reserved for the first iteration.

We tested three different values of γ, i.e., 0, 0.25, and 0.5. The results in terms of quality of the solutions obtained by Gurobi and by our IKS are shown in Tables 1-3 regarding $|K| = 5$ instances, and in Tables 4-6 regarding $|K| = 10$ instances. For each instance, identified by the tuple $\{|M|, |F|, \theta\}$, we report:

- *gap*: the percentage distance between the best integer solution and the best bound (for Gurobi only);
- *obj*: the best objective function value;
- *sat*: the percentage of totally satisfied patients.

The symbol "-" is used if no solution has been found within the time limit. We also report, for both Gurobi and IKS, the average *obj* and *sat* values. Moreover, the number of times in which Gurobi and IKS have obtained the best solution and the best satisfaction percentage is reported in the last row (*#best*). Finally, for each entry, the best *obj* and *sat* values between the two methods are highlighted in bold font.

If we consider the instances with $|K| = 5$ (Tables 1-3), IKS performs better than Gurobi, despite having a shorter time limit. In fact, IKS obtains the best solution 35 times, while Gurobi only 28. On average, IKS also obtains a better objective function value (or very similar in Table 3), which indicates a higher performance consistency. Instead, analyzing the *sat* value is not a straightforward task. In fact, there are several cases in which having a higher *sat* value does not correspond to a higher *obj* value. A case in point is the fact that Gurobi shows a higher *sat* average over the instances in these tables, but not a higher *obj* value average. It is worth noticing that, as expected, the higher the bonus γ, the higher the *sat* average. For example, IKS goes from a 30.3% *sat* average when γ is 0, to a 39.6% *sat* average when γ is 0.5.

[1] http://www.gurobi.com/. Last access on Jan 05, 2019.

If we consider the results presented in Tables 4-6 for $|K| = 10$, the comparison still favors IKS. Gurobi obtains 28 best results, while IKS 32. IKS also obtains by far the best obj average, mainly due to the fact that Gurobi is unable to find any feasible solution in several occasions (7 times out of 60). The upwards trend in the satisfaction percentage is confirmed, although the average objective function value does not increase as much as expected, when moving from a zero bonus situation to a 50% bonus situation.

In Table 7 we make a detailed comparison between the total computational times (t) needed by the two methods, and their time-to-best (ttb), i.e. the time needed to achieve the best solution. Each entry reports average values among

Table 1. Instances with $|K|$=5 and $\gamma = 0$.

Instance			**Gurobi**			**IKS**	
$\lvert M\rvert$	$\lvert F\rvert$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	2	5.1	**5672**	**88**	5481	83
25	3	4	24.9	4825	**21**	**5221**	**21**
25	5	2	0.0	**4921**	**100**	**4921**	**100**
25	5	4	26.3	6078	54	**6975**	**63**
50	3	2	14.6	**4530**	**27**	4464	22
50	3	4	26.3	**7396**	**22**	7210	8
50	5	2	8.1	**6369**	**39**	6361	**39**
50	5	4	55.8	6200	**29**	**6880**	18
75	3	2	3.7	8811	**24**	**8931**	**24**
75	3	4	57.6	6167	**11**	**9374**	8
75	5	2	25.9	8428	**24**	**9228**	**24**
75	5	4	59.7	7365	12	**10936**	**15**
75	10	2	26.4	**13244**	**47**	12557	41
75	10	4	12.6	17426	**49**	**18167**	45
100	3	2	5.9	8004	**15**	**8009**	13
100	3	4	60.3	4703	**5**	**6556**	4
100	5	2	6.6	**11826**	**33**	11240	26
100	5	4	41.6	8807	**8**	**9806**	**8**
100	10	2	42.7	10288	30	**12970**	**33**
100	10	4	12.5	**21431**	**19**	16839	11
avg.:				8624.6	**33.8**	**9106.3**	30.3
#best:				8	**17**	**13**	9

Table 2. Instances with $|K|$=5 and $\gamma = 0.25$.

Instance			**Gurobi**			**IKS**	
$\lvert M\rvert$	$\lvert F\rvert$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	2	10.7	6671.00	83	**7253.75**	**96**
25	3	4	13.2	**6534.50**	**42**	6035.50	**42**
25	5	2	0.0	**6151.25**	**100**	**6151.25**	**100**
25	5	4	34.4	6760.25	**58**	**7898.00**	50
50	3	2	11.6	**5124.75**	**33**	4842.75	27
50	3	4	20.5	**9034.75**	**29**	7804.50	16
50	5	2	10.0	7296.00	43	**7481.75**	**47**
50	5	4	58.5	7171.00	**33**	**7621.00**	31
75	3	2	2.7	**9718.75**	**34**	9675.25	32
75	3	4	56.4	6941.75	**20**	**8583.25**	15
75	5	2	24.9	**10325.50**	**34**	9387.50	27
75	5	4	66.1	6859.00	12	**10377.00**	**22**
75	10	2	27.2	**15478.50**	**49**	15124.00	47
75	10	4	19.0	20015.50	**53**	**20023.25**	46
100	3	2	6.0	**8695.75**	**18**	8077.50	**18**
100	3	4	57.2	5201.75	5	**6703.00**	**11**
100	5	2	5.2	**13387.75**	**40**	12085.00	34
100	5	4	47.5	8384.00	12	**10708.50**	**16**
100	10	2	41.6	12574.50	34	**14718.75**	**35**
100	10	4	11.9	**23339.00**	**29**	16853.50	22
avg.:				9783.3	**38.1**	**9870.0**	36.7
#best:				10	**14**	**11**	9

all the instances tested with a specific combination of parameter γ and number of patients $|M|$. On the one hand, it is easy to see that IKS is able to obtain its best solution within 8 minutes in average, and it is fairly consistent across all the combinations of parameters. Gurobi, on the other hand, requires, on average, at least 4 times the time needed by IKS to obtain its best solution. If we consider that IKS obtains, on average, solutions that are better than the one of Gurobi, this clearly shows how IKS is the best performing method out of the two.

6. CONCLUSIONS

Home health-care is a fast growing industry and, considering the progressive aging of the European population (Eurostat European Commission, 2008), the growth rate is only destined to increase. In this setting, health-care providers have to manage an increasing number of requests, while the availability of human resources may not follow the same trend. In this context, we propose a variant of the NRP that addresses the problem of determining which services can be performed in a working day, while considering additional factors different from the pure profit. Our proposed variant, called PSNRP-TW, takes into account the need to balance profit and patient satisfaction. We proposed an Iterative Kernel Search algorithm to tackle the problem, and showed that its performance are consistent across different types of instances, and, on average, better than the one of a state of the art MILP solver, even when a shorter time limit is used.

Table 3. Instances with $|K|$=5 and $\gamma = 0.5$.

Instance			**Gurobi**			**IKS**	
$\|M\|$	$\|F\|$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	2	23.9	6824.5	71	**8583.0**	**92**
25	3	4	19.9	**7166.5**	**38**	6096.0	33
25	5	2	0.0	**7381.5**	**100**	**7381.5**	**100**
25	5	4	31.4	8481.5	54	**9058.5**	**58**
50	3	2	12.2	5729.5	**35**	**5796.5**	33
50	3	4	20.0	**10598.0**	**35**	8964.0	24
50	5	2	14.1	**8120.5**	41	7970.0	**43**
50	5	4	64.8	7283.0	29	**8739.0**	**37**
75	3	2	2.3	10894.0	35	**10902.5**	**36**
75	3	4	53.9	8063.5	16	**9401.0**	**19**
75	5	2	17.6	**12653.5**	**38**	10885.0	**38**
75	5	4	50.2	11053.0	21	**11278.5**	**22**
75	10	2	31.2	**17192.5**	**47**	16014.0	**47**
75	10	4	23.9	**22432.0**	**50**	20666.5	42
100	3	2	5.8	**9806.5**	26	9684.0	**29**
100	3	4	60.6	4913.5	5	**7527.0**	**11**
100	5	2	9.1	**14492.5**	**46**	14374.0	40
100	5	4	44.5	9466.5	9	**11981.5**	**17**
100	10	2	41.5	14910.5	**37**	**15838.0**	36
100	10	4	21.7	**24263.0**	31	19887.5	**34**
		avg.:		**11086.3**	38.2	11051.1	**39.6**
		#best:		10	9	**11**	**14**

Table 4. Instances with $|K|$=10 and $\gamma = 0$.

Instance			**Gurobi**			**IKS**	
$\|M\|$	$\|F\|$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	3	7.0	**7432**	**25**	7123	13
25	3	5	6.4	**7779**	**4**	7712	**4**
25	5	3	11.1	8306	**46**	**8366**	42
25	5	5	8.5	**9170**	21	8704	**25**
50	3	3	1.6	**7825**	10	7786	**12**
50	3	5	20.2	**13082**	**6**	11545	2
50	5	3	24.6	**8575**	**12**	8372	8
50	5	5	20.8	**19479**	**12**	18510	10
75	3	3	3.4	7634	**4**	**7635**	**4**
75	3	5	13.8	**12765**	**1**	11721	1
75	5	3	4.9	**10551**	**4**	9812	**4**
75	5	5	61.9	7391	1	**9832**	**5**
75	10	3	27.8	12782	5	**14498**	**7**
75	10	5	52.9	12982	**10**	**16612**	5
100	3	3	82.8	1384	**1**	**4268**	**1**
100	3	5	-	-	-	**15403**	**1**
100	5	3	9.6	**9843**	**7**	9034	6
100	5	5	61.6	7185	**4**	**8923**	**4**
100	10	3	43.6	12883	5	**13272**	**7**
100	10	5	-	-	-	**19346**	**1**
		avg.:		8852.40	**8.9**	**10923.7**	8.1
		#best:		**10**	**13**	**10**	12

Table 5. Instances with $|K|$=10 and $\gamma = 0.25$.

Instance			**Gurobi**			**IKS**	
$\|M\|$	$\|F\|$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	3	10.9	7878.50	**29**	**7899.5**	**29**
25	3	5	12.5	7826.25	**17**	**7887.25**	**17**
25	5	3	13.8	**9911.75**	**63**	9457.00	54
25	5	5	18.2	8473.25	29	**9960.50**	**38**
50	3	3	3.9	8037.50	14	**8050.25**	**16**
50	3	5	19.3	**13809.25**	**12**	12772.50	8
50	5	3	29.0	8563.00	**16**	**8869.75**	12
50	5	5	24.1	**19834.75**	**16**	19197.25	8
75	3	3	2.9	7865.75	**9**	**7878.75**	**9**
75	3	5	20.8	11812.25	**1**	**11995.00**	0
75	5	3	10.2	**10289.25**	**8**	9769.00	**8**
75	5	5	54.5	9103.50	**5**	**11837.00**	4
75	10	3	23.3	**14587.25**	**12**	13563.50	**12**
75	10	5	37.9	17802.75	**12**	**18097.50**	7
100	3	3	85.8	1265.50	1	**3437.25**	**2**
100	3	5	-	-	-	**15265.25**	**2**
100	5	3	31.8	9513.50	11	**9911.75**	**12**
100	5	5	45.1	**10569.00**	**5**	8859.75	3
100	10	3	27.0	**17281.75**	**15**	13305.25	8
100	10	5	-	-	-	**17584.75**	**4**
		avg.:		9721.3	**13.8**	**11279.9**	12.7
		#best:		**7**	**14**	**13**	11

Table 6. Instances with $|K|$=10 and $\gamma = 0.50$.

Instance			**Gurobi**			**IKS**	
$\|M\|$	$\|F\|$	θ	gap(%)	obj	sat(%)	obj	sat(%)
25	3	3	16.6	8431.0	29	**9255.0**	**42**
25	3	5	9.7	8990.5	**29**	**9057.5**	**29**
25	5	3	16.2	**11484.0**	**63**	10089.5	54
25	5	5	18.7	**10986.5**	**33**	10837.0	**33**
50	3	3	4.3	8454.5	16	**8497.5**	**20**
50	3	5	25.4	**13580.0**	**10**	11378.0	6
50	5	3	10.6	**11868.0**	**26**	9516.0	16
50	5	5	27.5	**20342.5**	**12**	19603.0	**12**
75	3	3	3.3	**8502.5**	12	8499.50	**14**
75	3	5	15.2	**12791.5**	**1**	11886.0	0
75	5	3	10.6	**11341.0**	**15**	10320.5	**15**
75	5	5	47.6	**10903.5**	**4**	10866.5	**4**
75	10	3	20.2	**17041.5**	**23**	15626.0	20
75	10	5	57.0	13260.5	11	**19017.5**	**15**
100	3	3	93.2	678.0	0	**5337.5**	**6**
100	3	5	-	-	-	**15144.5**	**4**
100	5	3	-	-	-	**10668.0**	**18**
100	5	5	73.7	7362.0	**6**	**8996.0**	3
100	10	3	36.7	**15906.0**	**13**	14341.0	12
100	10	5	-	-	-	**25395.0**	**4**
		avg.:		9596.2	15.5	**12216.6**	**16.4**
		#best:		**11**	**12**	9	**13**

As future developments, we might make the problem more realistic introducing complicating features such as skill or capacity restrictions. Moreover, we might study the problem in a multi-period setting (e.g., planning each day of a week) or considering uncertain parameters (e.g., traveling and service times).

Table 7. Averages of computational times and time-to-best for different values of $|M|$ and γ.

		t(s)		ttb(s)	
$\|M\|$	γ	Gurobi	IKS	Gurobi	IKS
25	0.00	3180	600	1995	454
25	0.25	3170	600	2152	449
25	0.50	3262	600	2094	441
	avg.:	3204	600	2080	448
50	0.00	3600	600	2690	484
50	0.25	3600	600	1885	536
50	0.50	3600	600	2097	484
	avg.:	3600	600	2224	501
75	0.00	3600	600	2167	484
75	0.25	3600	600	2311	460
75	0.50	3600	600	2049	436
	avg.:	3600	600	2176	460
100	0.00	3600	600	2242	444
100	0.25	3600	600	2378	383
100	0.50	3600	600	2367	397
	avg.:	3600	600	2329	408
global avg.:		3501	600	2202.2	450

REFERENCES

Angelelli, E., Mansini, R., and Speranza, M.G. (2010). Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11), 2017 – 2026. Metaheuristics for Logistics and Vehicle Routing.

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

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

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

Chahed, S., Marcon, E., Sahin, E., Feillet, D., and 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.

Eurostat European Commission (2008). Population projections 2008-2060. `http://europa.eu/rapid/press-release_STAT-08-119_en.htm`, Last accessed on 26-02-2019.

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

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

Gendreau, M., Manerba, D., and 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.

Hanafi, S., Mansini, R., and Zanotti, R. (2018). Multi-visit clustered team orienteering problem. *submitted*.

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

Lin, C.C., Hung, L.P., Liu, W.Y., and Tsai, M.C. (2018). Jointly rostering, routing, and rerostering for home health care services: A harmony search approach with genetic, saturation, inheritance, and immigrant schemes. *Computers & Industrial Engineering*, 115, 151–166.

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

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

Manerba, D. and Mansini, R. (2016). The nurse routing problem with workload constraints and incompatible services. *IFAC-PapersOnLine*, 49(12), 1192–1197. doi:http://dx.doi.org/10.1016/j.ifacol.2016.07.670. 8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM) Troyes, France, 28–30 June 2016.

Milburn, A.B. (2012). Operations research applications in home healthcare. In R. Hall (ed.), *Handbook of Healthcare System Scheduling*, volume 168 of *International Series in Operations Research & Management Science*, 281–302. Springer US.

Nasir, J.A. and 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.

Ozcan, Y.A. (2008). Nursing home applications. In *Health Care Benchmarking and Performance Evaluation*, volume 120 of *International Series in Operations Research & Management Science*, 143–153. Springer US.

Vansteenwegen, P., Souffriau, W., Berghe, G.V., and Oudheusden, D.V. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12), 3281–3290.

Yang, X., Zhang, T., Mao, N., Chen, Q., Bai, L., and Chen, W. (2018). The stochastic home health care routing problem in dense communities with multi-appointment. In *2018 15th International Conference on Service Systems and Service Management, ICSSSM 2018*.