Optimizing Staff Scheduling in Emergency Medical Services: a case at INEM

Joana Maria Martins Namorado Rosa

Paper to obtain the Master of Science Degree in Biomedical Engineering
Instituto Superior Técnico

Abstract - The staff scheduling problem involves assigning people to tasks, integrated in working shifts. It is a complex and time-consuming activity common to several real-world companies. Quite often, staff scheduling problems comprise conflicting objectives. These problems are usually conditioned by legal and working rules, and by personal's preferences. Thus, the challenge is not simply attaining feasible schedules but further finding schedules that most accurately fit staff demands, cost savings, individuals' satisfaction, equity aspects, etc.

For that purpose, a standard integer programming formulation is proposed, and a general column generation-based diving heuristic approach is developed to solve the scheduling problem. The model is generic and possibly adjusted to several realities and companies. In this context, the model is applied to solve a real-life problem at Instituto Nacional de Emergência Médica (INEM). The quality and the computational time of the solutions obtained are analyzed, demonstrating that the heuristic developed finds good quality solutions for real instances in relatively short running times. The best-found solution is compared with a planned schedule of INEM, strengthening the practical value of the model. Since the schedules at INEM are still manually performed, an automatic generator scheduling application to produce solutions with an intuitive graphical user interface is implemented. The ultimate goal is to use the tool to support INEM in their staff scheduling activities while enhancing finance viability, by detecting current scheduling-related issues and by constantly improving schedules' quality.

Keywords - Staff Scheduling, Emergency Medical Services, Optimization, Column generation, Diving Heuristic

I. Introduction

Staff scheduling is identified as the process of deploying timetables for a set of workers within an organization so as to satisfy demand for various services, while simultaneously ensuring a distinctive level of employee satisfaction (Ernst et al., 2004b). Moreover, staff scheduling also needs to consider legal, organizational and contractual constraints (Van Den Bergh et al., 2013). In emergency medical institutions, staff scheduling is of paramount importance, since shortages in the number of required personnel directly impacts on the quality of care that patients receive. Furthermore, employee satisfaction cannot be neglected, as undesirable schedules can lead to increased staff turnover and to poorer on-job-performances (Cline et al., 2003).

Regardless of the high complexity in building rosters, several institutions still plan their timetables manually, requiring a lot of administrative work. In parallel, it is not easy for human planners to take all the different constraints and objectives into account. By contrast, an automated scheduling tool may provide solutions in significantly less time, while improving the solution's quality. Transparency would also be enhanced as the rules an algorithm uses are agreed beforehand, increasing employee perception about fairness and equity.

The present master paper addresses a staff scheduling problem motivated by the real-life context at *Instituto Nacional de Emergência Médica* (INEM), the main emergency medical institution in Portugal. The study has a twofold motivation. It intends, from an academic perspective, to be innovative and enrich the literature, by developing a novel model that may be applied to other real-life settings. From INEM view, it aims to enhance INEM productivity and profitability levels, by noticing and proposing solutions for their current scheduling conflicts.

II. Literature Review

Staff scheduling problems are common for most organizations in a wide variety of settings, and has been extensively studied in the literature, however far from achieving consensus or ultimate responses (Ernst et al., 2004a). Causmaecker et al. (2005) propose a classification for these problems based into four categories: (1) permanence centered planning, consisting of problems where the volume of personnel required is known in advance; (2) in fluctuation centered planning, demand changes throughout the day, such as in call centers or fast food restaurants; (3) mobility centered planning, the case of e.g. railway and airline companies, where tasks involve transportation of workers from one place to another; (4) finally, in areas such as consultancy, work is typically divided into projects to which different groups of employees are assigned, and this belongs to the project centered planning category. The present problem at INEM is a mix of the first three categories, as the required number of personnel for each task is known *a priori*, but differs between morning, afternoon, or night shifts, and different teams operate in different locations. Additionally, it is worth mentioning an annotated scheduling bibliography of ca. 700 articles with a short summary of each paper presented by

Ernst et al., 2004b.The papers are classified according to the type of problem addressed, the application areas covered, and the methods used. Only some of these papers tackle the EMS field.

Several solution techniques have been employed to solve personnel scheduling problems. A board classification divides these techniques into exact and heuristic. A straightforward approach is to formulate the problem as an integer programming model and solve it to optimality using a general IP solver (Isken, 2004). There are other relevant exact solution approaches such as branch-and-price, which uses Column Generation (CG) in each node of the branch-and-bound tree to solve the LP relaxation (Hans , 2001). After solving the master problem, it is necessary to check the optimality of the LP solution. To check the solution, it is solved a subproblem known as pricing problem that identifies the columns that should be considered. If the solution of the pricing problem is not feasible, there is at least one violated constraint (i.e. at least one column with negative Reduced Cost - RC) and these columns are added to the master, which is reoptimized. The scheme terminates when the LP relaxation is totally solved, which happens when there are no more columns with negative RC (Hans , 2001). Generally, after solving the CG scheme, the solution arising is fractional. Branching is then performed to find a feasible integer solution to the IP problem. Vanderbeck (2000) addresses some branching schemes and their respective implications on the CG scheme.

Despite the vast improvements in computer hardware and commercial IP solvers in the last decades, staff scheduling problems remain difficult to solve to optimality. Furthermore, optimal solutions that require many hours to calculate are often less valuable than quick suboptimal solutions, which allow user feedback or sensitivity analysis. Also, heuristics are relatively easy to implement and are able to deal with complex constraints or objectives. This led researchers to incorporate exact and heuristic methods using hybrid methods (Cheang, Li, Lim, & Rodrigues, 2003). Indeed, CG can be used inside a heuristic framework (Joncour et al., 2010; Gomes et al., 2017). Gomes et al. (2017) combine a CG approach through an IP formulation within a VNS heuristic to efficiently find columns with negative RC. Alternatively, CG can be combined with other heuristics such as a diving heuristic, proposed by Joncour et al. (2010). Diving heuristics are used to obtain integer feasible solutions. The method heuristically selects a branch in the branch-and-price tree, by using a certain rounding strategy such as rounding down, up, to the closest integer, or based on a threshold. Quite often the nodes are explored by fixing one or more columns to one so that the remaining solution space is reduced much more, therefore significantly speeding up the process. After each branching decision, new columns are generated for the people for which no column has been fixed yet (Joncour et al., 2010).

Finally, many different models having been proposed in the literature to solve personnel scheduling problems, only around 30% are implemented and applied in practice (Kellogg and Walczak, 2007). This study fills the implementation gap, by developing a graphical interface that allows the implementation of the proposed algorithm and an easy communication of the outcomes.

III. Model Formulation

a. Problem Statement

The present problem jointly considers the staffing of a set of services. Each service operates 24/7. Each day of the planning horizon is divided into three shifts in which tasks are performed: a night shift (N) from midnight to 8 am, a morning shift (M) from 8 am to 4 pm, and an afternoon shift (A) from 4 pm to midnight. The model allows that some tasks have duration that differs from the shift length. The task then starts at the same time as the shift it is assigned to, but can finish either before or after the end of the shift. All staff members are assigned to one of the existing services. Each service is divided in a number of teams. To these teams belong both a set of tasks and a set of people. Workers may belong to one or more than one team. Preferably workers perform tasks in their own team(s), but it is possible to do tasks from other teams/services to meet the required demand. However, this should be avoided if possible.

The primary objective of the schedule is to ensure functionality of the services. Workforce demands are determined in advance and can vary between shifts. In addition, legal rules require a minimum resting time of two shifts between each pair of shifts worked. Working time regulations limit the maximum number of consecutive working days and of consecutive days off for each person. Workers must have a minimum number of Sundays off over the planning period. Furthermore, is necessary to guarantee that employees are only assigned to tasks which they are able to perform. On the other hand, the schedules should aspire to be equitable to motivate and engage the staff. For that, every person needs to work at least a foretell number of night, morning and afternoon shifts. Workers should have the entire weekend off instead of a single day. Each worker's contract hours should be met as much as possible, meaning both overtime and undertime are undesirable. Finally, the number of tasks assigned to people from other teams should be minimized. These last three constraints are soft constraints, while the others are hard constraints. Hard constraints must be satisfied to obtain a feasible solution, while the soft ones should be satisfied, as much as possible.

b. Standard IP Formulation

Notation for the standard IP is first introduced, including sets, subsets, parameters and decision variables.

- Sets
  - $i \in I$: the set of people
  - $t \in T$: the set of tasks
  - $d \in D$: the set of days in the planning horizon
  - $w \in W$: the set of full weekends in the planning horizon
  - $s \in S$: the set of shifts, i.e. S = {night, morning, afternoon}
  - $g \in G$: the set of working teams
  - $j \in J$: the set of working services

- Subsets
  - I_t^T: the subset of people that can perform task *t*
  - I_g^G: the subset of people that belong to team *g*
  - T_g^G: the subset of tasks that belong to team *g*
  - T_j^J: the subset of tasks belonging to service *j*
  - G_j: the subset of teams belonging to service *j*

- Parameters
  - k: starting weekday of the planning horizon (0 = Monday, 1 = Tuesday, …, 6 = Sunday)
  - ξ: number of holidays of the planning horizon
  - θ_i: number of contract hours of person *i*
  - η: number of hours to discount from the contract hours θ_i per holiday ξ
  - L_t: duration of task *t*
  - R_{tds}: required number of people to be assigned to task *t* on shift *s* and day *d*
  - θ^1: maximum number of consecutive working days
  - θ^2: maximum number of consecutive days off
  - θ^3: minimum number of Sundays off
  - θ_s^4: minimum number of shifts of type *s* necessary to be worked
  - w_j^{RE+}, w_j^{RE-}: weight of penalty variables for excess and shortage workforce supply in service *j*, respectively
  - w^{WO}: weight of penalty variable for full weekend off
  - w^{H+}, w^{H-}: weight of penalty variable for excess and shortage hours worked, respectively
  - w_j^G: weight of penalty variables for assigning tasks of a team to members of another team in service *j*

- Decision Variables
  - $x_{itds} \in \{0,1\}$: equals 1 if person *i* is assigned to task *t* on shift *s* and day *d*, 0 otherwise
  - $Y_{tds}^{RE+}, Y_{tds}^{RE-} \in \mathbb{N}$o: penalty variables for excess and shortage of workforce supply for task *t* on shift *s* and day *d*, respectively
  - $Y_{iw}^{WO+}, Y_{iw}^{WO-} \in \mathbb{N}$o: penalty variables for full weekend off for person *i* in weekend *w*
  - $Y_i^{H+}, Y_i^{H-} \in \mathbb{N}$o: penalty variable for excess and shortage hours worked for person *i*, respectively
  - $Y_g^G \in \mathbb{N}$o: penalty variables for assigning tasks of a team *g* to members of another team

- Objective Function

$$\text{minimize: } \sum_{d\in D}\sum_{s\in S}\sum_{j\in J}\sum_{T_j^J}\left(w_j^{RE+}Y_{tds}^{RE+} + w_j^{RE-}Y_{tds}^{RE-}\right) + \sum_{i\in I}\sum_{w\in W} w^{WO}\left(Y_{iw}^{WO+} + Y_{iw}^{WO-}\right) + \sum_{i\in I}\left(w^{H+}Y_i^{H+} + w^{H-}Y_i^{H-}\right) + \sum_{j\in J}\sum_{g\in G_j}\left(w_j^G Y_g^G\right) \quad (1)$$

- Constraints

$$\sum_{i\in I_t^T} x_{itds} - Y_{tds}^{RE+} + Y_{tds}^{RE-} = R_{tds}, \qquad \forall\, t \in \mathrm{T}, \mathrm{d} \in \mathrm{D}, \mathrm{s} \in \mathrm{S} \tag{2}$$

$$\sum_{\mathrm{t}\in\mathrm{T}} (\mathrm{x}_{\mathrm{itd,night}} + \mathrm{x}_{\mathrm{itd,morning}} + \mathrm{x}_{\mathrm{itd,afternoon}}) \leq 1, \qquad \forall \mathrm{i} \in \mathrm{I}, \mathrm{d} \in \mathrm{D} \tag{3}$$

$$\sum_{\mathrm{t}\in\mathrm{T}} (\mathrm{x}_{\mathrm{itd,morning}} + \mathrm{x}_{\mathrm{itd,afternoon}} + \mathrm{x}_{\mathrm{it,d+1,night}}) \leq 1, \qquad \forall \mathrm{i} \in \mathrm{I}, \mathrm{d} \in \mathrm{D}\backslash\{|\mathrm{D}|\} \tag{4}$$

$$\sum_{\mathrm{t}\in\mathrm{T}} (\mathrm{x}_{\mathrm{itd,afternoon}} + \mathrm{x}_{\mathrm{it,d+1,night}} + \mathrm{x}_{\mathrm{it,d+1,morning}}) \leq 1, \qquad \forall \mathrm{i} \in \mathrm{I}, \mathrm{d} \in \mathrm{D}\backslash\{|\mathrm{D}|\} \tag{5}$$

$$x_{itds} = 0, \qquad \forall t \in T, i \in I\backslash I_t^T, d \in D, s \in S \tag{6}$$

$$\sum_{t\in T} \sum_{r\in\{d,d+1,\dots,d+\theta^1\}} \sum_{s\in S} x_{itrs} \leq \theta^1, \qquad \forall i \in I, d \in D \ \backslash\{|D|, |D|-1, \dots, |D|-\theta^1+1\} \tag{7}$$

$$\sum_{t\in T} \sum_{r\in\{d,d+1,\dots,d+\theta^2\}} \sum_{s\in S} x_{itrs} \geq 1, \qquad \forall i \in I, d \in D \ \backslash\{|D|, |D|-1, \dots, |D|-\theta^2+1\} \tag{8}$$

$$\sum_{t\in T} \sum_{d\in\{7-k,\ 7-k+7,\dots\}} \sum_{s\in S} x_{itds} \leq |W| - \theta^3, \qquad \forall i \in I \tag{9}$$

$$\sum_{t\in T} \sum_{d\in D} x_{itds} \geq \theta_s^4, \qquad \forall i \in I, \forall s \in S \tag{10}$$

$$\sum_{t\in T} \sum_{s\in S} (x_{it,7-k+7(w-1),s} - x_{it,6-k+7(w-1),s}) - Y_{iw}^{WO+} + Y_{iw}^{WO-} = 0, \qquad \forall i \in I, w \in W \tag{11}$$

$$\sum_{t\in T} \sum_{d\in D} \sum_{s\in S} L_t x_{itds} - Y_i^{H+} + Y_i^{H-} = \theta_i - \eta\xi, \qquad \forall i \in I \tag{12}$$

$$\sum_{i\in I_g^G} \sum_{t\in T\backslash T_g^G} \sum_{d\in D} \sum_{s\in S} x_{itds} - Y_g^G = 0, \qquad \forall g \in G \tag{13}$$

The problem soft constraints are weighted in the objective function (1). The overall objective is to minimize the weighted sum of these penalty variables, thus minimizing the real impact of violating the soft constraints. Constraint (2) is the coverage requirement. Accordingly, staff must have a certain number of resting hours between consecutive shifts, which is enforced by constraints (3), (4), (5) for night, morning, and afternoon shifts, respectively. Staff cannot be assigned to tasks which they cannot perform (6). Staff are not allowed to work more than θ^1 days consecutively (7) and they cannot have θ^2 or more consecutive days off (8). In every planning period, each person must have at least θ^3 Sundays off (9). Each person needs to work at least θ_s^4 shifts of each type s (10). Preferably, people get the entire weekend off instead of a single day (11). Ideally, people work their specified number of contract hours (θ_i), adjusted for the number of holidays (ξ) in the planning period, properly pondered (η) (12). Finally, tasks belonging to a certain group should be assigned to members of that group (13).

c. Alternative Hybrid Formulation

The integer model consisting of (1) - (13) can be formulated in a different way. For every person, a work pattern can be defined as the tasks assigned to that person over the planning horizon. This activity pattern, which corresponds to certain decision variables of the problem, is referred to as a column. This approach requires a revision of the notation and readjustments on the original IP formulation.

- Revised notation
  - k_i: set of columns for person i
  - a_{iktds} : equals 1 if column *k* for person *i* assigns task *t* on shift *s* of day *d*, 0 otherwise
  - c_{ik}: the cost of column *k* for person *i*

- Decision variables
  - $z_{ik} \in \{0,1\}$: equals 1 if column *k* is chosen for person *i*, 0 otherwise

- Master Problem

$$\text{minimize: } \sum_{i\in I}\sum_{k\in K_i}(c_{ik}z_{ik}) + \sum_{d\in D}\sum_{s\in S}\sum_{j\in J}\sum_{t\in T_i^J}(w_j^{RE+}Y_{tds}^{RE+} + w_j^{RE-}Y_{tds}^{RE-}) \quad (14)$$

$$\sum_{i\in I}\sum_{k\in K_i} a_{iktds}z_{ik} - Y_{tds}^{RE+} + Y_{tds}^{RE-} = R_{tds}, \qquad \forall t \in T, d \in D, s \in S \quad (15)$$

$$\sum_{k\in K_i} z_{ik} = 1, \qquad \forall i \in I \quad (16)$$

$$c_{ik} - \sum_{t\in T}\sum_{d\in D}\sum_{s\in S} a_{iktds}\lambda_{tds} - \mu_i \quad (17)$$

Equation (14) is the objective function of the master problem. The main goal is to minimize two terms: the costs of the chosen activity patterns for every person in terms of z_{ik}, and the under and overstaffing costs within each service *j*. The adapted master problem has constraints (15) – (17). Constraints (15) are the coverage requirements, (16) enforces that exactly one working pattern is chosen for each person.
Let λ_{tds} represent the dual costs associated with constraints (15) and μ_i the dual costs associated with constraints (16). The RC of a new column *k* for person *i* is then given by equation (4.17), where c_{ik} is the cost of the column based on the soft constraint violations.

- Subproblems

The decision variables for the pricing problem are now $a_{iktds} \in \{0,1\}$: equal 1 if task *t* is assigned on shift *s* of day *d*, otherwise it is 0. The pricing problem for each person *i* can then be defined as stated in (14). The set of constraints of the pricing problem consist on equations (3) – (13), but with the revised decision variables a_{iktds}, instead of the former x_{itds}.

$$\text{minimize: } \sum_{t\in T}\sum_{d\in D}\sum_{s\in S}(-a_{iktds}\lambda_{tds}) + \sum_{w\in W}(w^{WO}Y_w^{WO+} + w^{WO}Y_w^{W0-}) + w^{H+}Y^{H+} + w^{H-}Y^{H-} + \sum_{j\in J}\sum_{g\in G_j} w_j^G Y_g^G \quad (14)$$

d. Strategies to Solve the Hybrid Formulation

- CG initialization and loop

The master problem is initialized with 'supercolumns' with very high cost in which all entries are set to zero. This means that every entry of every column *k* for every person *i* is 0, irrespective of the task *t*, shift *s* and day *d*.
The CG loop can be implemented in different ways (see e.g. Beliën and Demeulemeester, 2006). In this study, two possibilities are considered for the loop implementation.
A first possibility is to solve the master problem, obtaining the dual variables for each constraint in the master, and then use this information to solve one subproblem for every person *i*, through objective function (18). All columns that are found with a negative RC are added to the master problem, after which a new iteration begins. In the new iteration, the master generates a new set of dual variables and the subproblems are solved again. A second possibility is to solve the master problem and then solve a subproblem for a single person *i*. Similarly, the columns with negative RC are added to the master, which is reoptimized. Then a subproblem is solved for person *i + 1* and so on. For both alternatives, once all subproblems have been solved and there are no columns with negative RC variables, the master is no longer repeated, and the problem is solved to optimality.

- Stopping Criteria of the CG phase

The perception of the tailing-off effect is extremely convenient to timely stop the CG scheme. In the tailing-off effect at first new columns improve the LP solution significantly, while in the later iterations each additional column only improves the objective function value very slightly. When the aim is to quickly find integer solutions, the CG phase can be terminated prematurely, and the branching started before LP optimality has been reached. Hence, the challenge is to stop generating columns as soon as this effect becomes perceptible, without compromising the LP solution.
In this paper, the CG scheme is stopped after a predetermined number of iterations. For this option, two positive integer parameters are set: β_{root} for the number of iterations in the root node and β_{diving} for the number of iteration in each

node during diving. The higher the value of these parameters, the more accurate the solution is expected to be but the more it takes until finishing the CG phase.

- Diving Heuristics

The solution found through CG might not be integral. Therefore, branching is needed to find an integer solution. However, as an exact branch-and-price approach requires a prohibitively large amount of computation time for large problem instances, a diving heuristic is used to quickly find good integer solutions. In a diving heuristic, the branch-and-price tree is traversed in a depth-first manner until finding a feasible solution.

The branching can be done in multiple ways. There are two ways considered in this paper. One way is to fix the largest fractional variable to one (i.e. the column with the highest variable). Another possibility is to fix all fractional variables with a value above a certain threshold δ to one. While in the first alternative, only one person can be removed per iteration, in the second alternative more than one person can be removed per cycle. Irrespective of the alternative, after each search in the diving tree, the person whose column has been fixed is eliminated from the problem. For the ones whose partial schedule has not yet been set, a new iteration is undertaken. The iterations proceed until one solution has been found for every person i considered in the problem.

IV. Application to the case-study at INEM

a. INEM context

The model previously formulated to is applied to the INEM context. This emergency institution operates on a continuous basis, 24 hours every day. It consists of two main services: the emergency dispatch centers known as *Centro de Orientação de Doentes Urgentes* (CODU) and the Emergency Vehicles (EVs). CODU is responsible for acquiring the calls, delivering voice assistance and instructions, and dispatching the convenient transportation, while EVs provide assistance and care to the patients. The scope of this study is the Lisbon region and some neighboring areas where INEM vehicles are located and are coordinated by the Lisbon center (i.e. Almada, Amadora, Cascais, Elvas, Estremoz, Ponte de Sor, Sacavém, Seixal, Setúbal, Tomar and Torres Novas.)

The workforce at INEM is mainly composed of *Técnicos de Emergência Pré-Hospitalar* (TEPHs), i.e. technical personnel. Each TEPH is allocated to one of the two services, CODU or EVs. TEPHs can sporadically perform tasks at the other service. Within their service, TEPHs belong to teams. Each TEPH in CODU belongs to a single team, whereas in the EVs may belong to more than one team. Which team TEPHs are assigned to depends on two main factors. First, a TEPH only performs tasks if he/she has the required skills, e.g. a license for motorcycles to drive an emergency motorcycle. Second, TEPHs' residence place must be considered in the attribution of tasks, e.g. people from Lisbon cannot be assigned to teams from Torres Novas, which is located more than 100 kilometers away.

The EVs consist of different types of ambulances, namely Medical Emergency Ambulances (AEM), Immediate Life-Support Ambulances (SIV), Inter-hospital Pediatric Transports (TIP), Mobile Units of Psychological Emergency Intervention (UMIPE), and Medical Emergency Motorcycles (MEM). Every type of ambulance requires one TEPH to operate a single shift, except AEMs that require two TEPHs. For both CODU and the EVs, there are 8 types of tasks in total, which are summarized in Table 1. Each task has a duration equal to the shift length of 8 hours, except for the MEM tasks, which have a duration of 12 hours.

Table 1 - Overview of the different types of tasks at INEM and their duration.

Service	CODU		EVs					
Tasks	CODU shift responsible	CODU task	AEM driver	AEM team responsible	SIV task	TIP task	UMIPE task	MEM task
Task duration	8 hours							12 hours

Currently, the scheduling at INEM is done through an on-line platform, that visually communicates the schedules, the so-called *Gestão de Horários* (GH). At first, all employee proposes a schedule for themselves. In doing this, they are expected to adhere to certain constraints, to make sure feasible schedules can be obtained. Then in a second step, the coordinator for each team collects the data from all personnel and checks for infeasibilities. Reassignments are made until all constraints are met. This process can take several iterations. Finally, in a third step, the coordinators of the different teams synthesize their respective schedules and again reassignments are made between teams to better satisfy overall demand. GH does not consider legal, work-related or personnel requirements. These must be checked by the user, i.e. first the regular employee and in a further phase the coordinator of the team.

b. Input Analysis

At INEM, working time regulations state that people cannot work more than 6 days consecutively ($\theta^1 = 6$), cannot have 5 or more consecutive days off ($\theta^2 = 5$), and must have at least 1 Sunday off every four weeks ($\theta^3 = 1$). Additionally, it is required that each person works at least 2 shifts of each type, i.e. $\theta_s^4 = 2, \forall s \in S$. Finally, a standard contract specifies a working time of 140 hours per month and the number of holidays in the instances tested is zero, implying that $\theta_i = 140$ and $\xi = 0$, $\forall i \in I$.
Understaffing in the EVs is considered more undesirable than understaffing in CODU, as CODU continues to operate without significant problems if there is a small shortage in personnel on a certain shift, while the understaffing of EVs may lead to the non-operationality of the vehicles, thus $w_{EV}^{RE-} > w_{CODU}^{RE-}$. Because EVs are located in different regions, assigning people to tasks outside their team (if allowed based on the distance of the location) is worse for the EVs than for CODU, i.e. $w_{EV}^{G} > w_{CODU}^{G}$. The weights used in the all the tests are summarized in Table 2.

Table 2 - Objective function weights used in the computational tests.

Parameter	$\mathbf{w_{CODU}^{RE+}}$	$\mathbf{w_{EV}^{RE+}}$	$\mathbf{w_{CODU}^{RE-}}$	$\mathbf{w_{EV}^{RE-}}$	$\mathbf{w^{WO}}$	$\mathbf{w^{H+}}$	$\mathbf{w^{H-}}$	$\mathbf{w_{CODU}^{G}}$	$\mathbf{w_{EV}^{G}}$
Weight	10	10	100	1000	10	1	1	10	20

The overall problem has 289 TEHPs who are divided into 22 teams (5 in CODU and 17 in the EVs), and need to be assigned to 61 different tasks (10 in CODU and 51 in the EVs). The planning horizon considered is four weeks (28 days). Over the entire planning horizon, a total of 4527 task-demands need to be filled in. The starting day is set to 0, meaning the scheduling starts on a Monday. These input information is introduced in matrices form and read by the algorithm. For that, five different matrices are built. People are assigned to teams and tasks in table *person x team* and *person x task*, respectively. Each team comprises a specific set of tasks (*team x task*), and each task has specific demands (*task x demands*) and lengths (*length x task*).
To be validated, the performance of the algorithm is tested on different problem dimensions. In fact, the most important issues influencing the complexity of the INEM problem are: the number of TEPHs, the distribution of their skills, i.e. the number of people that can perform each task (this impacts the symmetry of the problem), and the length of the planning horizon. The number of tasks is implicitly determined by the number of TEPHs and narrowly linked to the distribution of their skills. Hence, four different datasets were generated based on the INEM dataset. The TestMD instance is obtained by extending the planning horizon from 28 to 56 days. The same task demands are used for both months. In the TestMP instance the number of workers is increased to 417. This number is chosen so that each team's size increases by the same factor. The task demands are increased at the same rate as the number of workers. The TestLS dataset is constructed by splitting tasks into two or more tasks, and assuming that people that have the skills to do the old task can only do one of the new tasks, so that symmetry is reduced. All task demands are divided proportionally over the new tasks. Finally, the TestHS dataset is derived by assuming every worker can do every task, so that symmetry is maximized. All other parameters are kept unchanged. Together with the CODU and EVs datasets, this leads to 7 datasets in total, which are summarized in Table 3.

Table 3 - Summary of the instances used to perform the computational tests. INEM instance results from combining CODU and EVs. Test MD, TestMP, TestLS, TestHS are varians of the INEM instance, used to validate the algorithm.

Instance	**CODU**	**EVs**	**INEM**	**TestMD**	**TestMP**	**TestLS**	**TestHS**
# people	68	221	289	289	417	289	289
# teams	5	17	22	22	22	22	22
# tasks	10	51	61	61	61	103	61
# days	28	28	28	56	28	28	28
# variables	57,120	946,764	1,480,836	2,961,672	2,136,708	2,500,428	1,480,836

c. Computational results

Both the standard IP model and the diving heuristic approach outlined are used to solve the current problem at INEM. The algorithm is coded in C++14 and compiled with Microsoft Visual Studio 2015. The callable library of ILOG

CPLEX 12.6.2 is used as IP solver. All tests are executed on a PC with an Intel Core i5-5200U CPU of 2.20 GHz and 8 GB of RAM under the Windows 10 operating system.

- CPLEX

Computational experiments are performed to test the standard IP Standard Formulation on three instances (CODU, EVs and INEM; the characteristics of these instances are available in Table 3). The objective is to understand how CPLEX outperforms on the staff scheduling problem at INEM.

The optimization has been first solved for CODU dataset as this is the simplest one. For CODU, CPLEX is able to find an optimal solution very fast (i.e. 3.472 seconds). The best-found solution (BFS) has an objective value of 4196, while the lower bound is already, very close, at 3941. The LP solution is used as lower bound and it corresponds to the objective value of the LP relaxation of the root node. The optimality gap is 0% and in the optimal solution, all CODU tasks are met.

On the other hand, for EVs and INEM dataset, CPLEX fails in finding an optimal solution within 5 hours of search. Indeed, for INEM dataset, by the time CPLEX is stopped, the BFS has an objective value of 1,901,916. Furthermore, it takes approximately 159 seconds to solve the LP relaxation of the root node, obtaining a value of 37,021. In this case, it is still quite small when compared to the objective value. The LP relaxation is not a feasible solution to the original problem, since the variables' domain is relaxed, i.e. fractional variables are allowed. However, the value of the LP relaxation is used as a lower bound to check the BFS quality. The maximal optimality gap as a percentage of the LP relaxation solved by CPLEX for INEM dataset is 98.05%. Through these tests, it is concluded that the optimization approach, using CPLEX solver, is not capable to solve the model for large instances.

- CG and diving heuristic methods

Motivated by the limitations of CPLEX, the diving heuristic is used as an alternative to solve the personnel scheduling problem at INEM. For this purpose, different algorithm configurations on the CODU, EVs and INEM dataset are tested to understand their impact on the solutions obtained.

The following notation is used to refer to the different algorithm configurations: A/B/C-D, where A refers to the CG method, B to the diving heuristic method, and C-D to the stopping criteria of the CG method based on the values of β_{root} and β_{diving}, respectively. The CG method (A) has values: E if a pricing problem is solved for every person in each iteration or P if only a pricing problem for person i is solved in each iteration. The branching method during diving (B) has values: L if the branch is on the highest fractional variable or T if the branching is on all variables with a value above a threshold (δ). In all tests where branching is done on variables with a value above a certain threshold δ, it is set $\delta = 0.6$. The stopping criteria (C-D) is based on the fixed number of iterations allowed, respectively: β_{root} is the number of iterations allowed in the root node or β_{diving} is the number of iterations allowed after each branching decision. The results for INEM instance are summarized in Table 4.

From the results exhibited, several conclusions can be made. First, the CG scheme where a subproblem is solved for every person during each iteration (scheme E) is outperformed by the scheme where the master is reoptimized each time a column is added (scheme P). Second, branching on the largest fractional value (scheme L) is not feasible as the required computation time is too large, irrespective of the CG scheme and the number of tolerable iterations. The only exception is configuration E/L/2-2, but this has a very poor solution (gap of 98.27%). The reasoning is directly related to the size and low-level of symmetry of the input data. Third, it is more advantageous to increase the number of iterations in the root node than during diving, but it extends the running time of the algorithm. Finally, for the INEM dataset, the optimality gaps are obtaining by comparing the BFS to the LP relaxation of the IP formulation solved with CPLEX, which in the INEM dataset is 37,021. This is a consequence of not knowing an optimal solution and in fact this lower bound might be relatively weak. Hence, from Table 4, configuration P/T/10-1 is the one achieving a better gap at 47.94%, followed by configuration P/T/3-3 with 58.57% and P/T/2-2 with 60.04%.

Summing up, the results for the INEM instance suggest that the diving heuristics clearly outperforms the original IP formulation solved by CPLEX. Whilst through CPLEX no good feasible solution is found within the 5-hours set, the heuristic algorithm found a solution in very reasonable time, for several configurations.

Table 4 - Results for different algorithm configuration on INEM dataset. BFS denotes the objective value of the Best-Found Solution, 'Objective root' the objective value of the LP relaxation of the root node, and 'Gap' the maximal optimality gap in percent compared to the LP relaxation solved by CPLEX.

	CPLEX	**E/L/2-2**	**E/L/ 10-1**	**E/T/2-2**	**E/T/3-3**	**E/T/ 10-1**	**P/L/2-2**	**P/L/ 10-1**	**P/T/2-2**	**P/T/ 3-3**	**P/T/ 10-1**
BFS	1,901,916	2,137,340	-	2,137,340	189,732	204,992	-	-	92,640	89,368	71,112
Total Time (s)	>18,000	52	>18,000	35	488	434	>18,000	>18,000	287	1,592	7,852
Objective root	37,021	2,137,340	515,331	2,137,340	1,727,850	515,331	103,131	61,703	103,131	86,628	61,703
Root time (s)	159	48	224	32	58	224	58	1,765	58	130	1,765
Gap	98.05%	98.27%	-	98.27%	80.49%	81.94%	-	-	60.04%	58.57%	47.94%

- Test sets

The algorithm configurations are validated by testing it on the generated problem dimensions introduced in Table 3. The purpose is to explore whether the former observations on the relative performance of the different algorithm configurations hold for other problem instances as well.
The results yield that extending the planning horizon (TestMD) to two months instead of a single month increases the complexity of the problem, and therefore the algorithm performs poorly. However, an increase on the number of people (instance TestMP) has small impact on the difficulty of the problem. As expected a reduction on the problem's symmetry (TestLS), i.e. a reduction on the possibility of permutation of the solutions, increases the problem's complexity; whereas an increase on the symmetry of the problem data (TestHS) simplifies the scheduling problem. It is intuitive as the chance of making bad decisions in the early stages of the heuristic is smaller for the case where every TEPH has the required skills to do every task. Later, at the end of the algorithm the people that haven't been yet assigned, can be assigned to the free tasks still left, without major problems.
To conclude, the results have shown that the model can easily handle problems with more people and is robust for the level of symmetry.

d. Comparison with a planned schedule at INEM

The best solution found arising from configuration P/T/10-1 is compared with a planned scheduled provided by INEM, so that the consistency of the model is appraised. Both schedules share most of the features, but whenever required, indicators are normalized to ensure veracity of the comparisons. Some relevant indicators are exhibited in Table 5.

Table 5 - Comparison of some relevant indicators: INEM real case and best solution case on CODU, EVs and INEM.

Indicator	**INEM real case**			**Best solution case**		
	CODU	EVs	INEM	CODU	EVs	INEM
Average hours worked (% of contractual hours)	106.95%	82.77%	88.69%	107.82%	102.31%	103.61%
Average number of Sundays off per person	1.69	1.09	1.24	1.18	1.38	1.33
Average number of weekends off per person	0.96	1.01	0.99	0.67	0.96	0.90

Globally, for the INEM real case, TEPHs are working below their contractual hours, and therefore INEM is not taking advantage of its full capacity. It may be justified by flaws in the intern organization but also by other factors, such as formation and training activities, vacations, etc. INEM real case allows many understaffing situations due to lack of symmetry and to the desire of meeting more personal preferences. On the other hand, in the best solution situation, in average a TEPH exceeds his/her working hours by only 3.61%. Indeed, the solution case is exploring more the TEPHs contractual hours, and therefore overtime is being used to satisfy the demand, as much as possible.
Following, it is verified that the average number of Sundays-off is always greater than 1, which is in accordance with the instruction from hard constraint (9). The chance of having an entire weekend off is not mandatory and therefore falls below the former value. The INEM real case assigns greater importance to this personnel preference, while the solution case schedule seeks primarily to satisfy the pre-defined demand. The comparison showed that the proposed model has practical meaning.

e. Graphical User Interface

The developed algorithm is implemented in a scheduling tool with a GUI that aims at achieving three objectives. First, it allows easy data input. The imported data are listed in a tree view so that it can be easily validated by the user. Second, it visualizes the found solution in two different ways: a first tab shows the tasks assigned to each person over the planning horizon, while a second tab shows for each task the demand in each period and the supply. Third, both the objective function weights as well as the algorithm configurations (CG scheme, branching method etc.) can easily be changed in the settings menu.

V. Conclusions

In this paper, the staff scheduling model is primarily formulated as a standard IP and in shown that for real-life problem instances a state-of-the-art commercial solver fails in finding a reasonably good solution within an acceptable time frame for large dimensions instances. Therefore, it is developed a CG-based formulation that decomposes the problem on its staff members. It assigns each person to a column that corresponds to an activity pattern. The solution from CG might not be integer thus diving heuristics is used to find an integer solution. Two CG schemes as well as two branching methods have been implemented. Additionally, as the CG phase requires a prohibitively large amount of time to solve to optimality, branching is terminated prematurely after a fixed number of iterations.

The model developed is tested on the real-life context at INEM. Comprehensive computational tests show that, compared to the standard IP model, the alternative hybrid approach succeeds in finding good quality solutions in reasonable computation times. The heuristic is validated by testing it on different dimensions instances, generated by altering the INEM instance.

In general, the algorithm is able to find good quality solutions in reasonable computation times. For the INEM instance, the results demonstrated that the BFS is for configuration P/T/10-1, with a gap of 47.94%. When comparing this solution to a INEM planned schedule several conclusions can be draw. The solution case is exploring more the TEPHs contractual hours. It is concluded that a higher symmetry, enhances the flexibility to assign people to tasks, therefore leading to enhanced schedules. The INEM real case schedule provides on average more weekend-off but less Sundays off to its workers. All in all, the analysis shows that the solution from the best configuration has practical value.

The findings of this work can be consolidated for future research so that the model is upgraded in performance and applicability aspects. Alternative heuristics can be explored to solve the model, namely a combination of a constructive and a variable neighborhood search (VNS) method. All in all, the analysis carried provides valuable insights not only to improve the scheduling making process but also to optimize the utilization of human and material resources at INEM.

VI. References

Beliën, J., & Demeulemeester, E. (2006). Scheduling trainees at a hospital department using a branch-and-price approach. *European Journal of Operational Research*, *175*(1), 258–278.

Causmaecker, P. De, Demeester, P., Berghe, G. Vanden, & Verbeke, B. (2005). Analysis of real-world personnel scheduling problems. In *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, 18th-20th August 2004, Pittsburgh, PA, USA* (pp. 183–198).

Cheang, B., Li, H., Lim, A., & Rodrigues, B. (2003). Cheang et al 2003_Review.pdf. *European Journal of Operational Research*, *151*(3), 447–460.

Cline, D., Reilly, C., & Moore, J. F. (2003). What's behind RN turnover? *Nursing Management*, *34*(10), 50–53. https://doi.org/10.1097/00006247-200310000-00016

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*(1), 3–27.

Gomes, R. A. M., Toffolo, T. A. M., & Santos, H. G. (2017). Variable neighborhood search accelerated column generation for the nurse rostering problem. *Electronic Notes in Discrete Mathematics*, *58*, 31–38.

Hans, E. W. (2001). *Resource loading by branch-and-price techniques*.

Isken, M. W. (2004). An Implicit Tour Scheduling Model with Applications in Healthcare. *Annals of Operations Research*, *128*(1–4), 91–109.

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., & Vanderbeck, F. (2010). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, *36*(C), 695–702.

Kellogg, D. L., & Walczak, S. (2007). Nurse scheduling: From academia to implementation or not? *Interfaces*, *37*(4), 353–369.

Van Den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, *226*(3), 367–385.

Vanderbeck, F. (2000). On Dantzig-Wolfe Decomposition in Integer Programming and ways to Perform Branching in a Branch-and-Price Algorithm. *Operations Research*, *48*(1), 111–128.