Effective Adaptive Large Neighborhood Search for a firefighters timetabling problem

Mohamed-Amine Ouberkouk, Jean-Paul Boufflet, Aziz Moukrim

Effective Adaptive Large Neighborhood Search for a firefighters timetabling problem

Mohamed-Amine Ouberkouk[1*], Jean-Paul Boufflet[1] and Aziz Moukrim[1]

[1*]Université de Technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203, Compiègne Cedex, France.

*Corresponding author(s). E-mail(s): mohamed-amine.ouberkouk@hds.utc.fr;

Abstract

Every year, wildfires accentuated by global warming, cause economic and ecological losses, and often, human casualties. Increasing operational capacity of firefighter crews is of utmost importance to better face the forest fire period that yearly occurs. In this study, we investigate the real-world firefighters timetabling problem of the INFOCA institution, Andalusia (Spain). The main issue is to achieve maximum operational capability while taking into account work regulation constraints. This paper proposes an Integer Linear Programming (ILP) formulation that makes it feasible to solve small/medium instances to optimality. We put forward a matheuristic (ILPH) based on the ILP formulation, and we obtain solutions for larger instances. We propose an Adaptive Large Neighbourhood Search metaheuristic (ALNS) to obtain better results for larger instances and we use a version of the ILPH as one of the constructive methods. The ALNS obtains all the optimal solutions found by the ILP on small instances. It yields better solutions than the ILPH matheuristic on larger instances within shorter processing times. We report on experiments performed on datasets generated using real-world data of the INFOCA institution. The work was initiated as part of the GEO-SAFE project.*

Keywords: Timetabling, Firefighters, ILP, Matheuristic, Adaptive Large Neighborhood Search

* https://geosafe.lessonsonfire.eu/

1 Introduction

In this study, we consider the **F**ire**F**ighters **T**imetabling **P**roblem (FFTP) of the INFOCA institution in Andalusia (Spain) whose mission is to fight against forest fires. We are given a number of firefighter crews, types of shifts and demands for these daily working shifts over a planning horizon. The objective is to build a schedule for every crew of firefighters, hence a full timetable for the forest fire period where wildfires yearly occur. Unfortunately, today's planning horizon is set to be broadened as a result of global warming, further motivating the need for an effective planning solution.

The members of a crew use to face wildfires under extreme conditions. Mutual confidence is the keystone of a firefighter crew, each crew member has as his/her first responsibility the live of the other members. The members of a firefighter crew are stable over the forest fire period, they use to train together to strengthens the crew cohesion.

The firefighter crews can be assigned to six different types of shifts (e.g. helicopter work, night work, or work on demand). There are different time slots and durations. Moreover, we need to fairly assign rest days and additional compensation days, granted when a number of hours have been worked. The overall operational capacity has to be ensured while strictly respecting the minimum demands and the regulatory constraints imposed by the institution. These relate to forbidden shift successions, maximum workload over the planning period, compensation days to be granted and maximum number of consecutive working days. Constraints of good practice have also to be considered to make the timetable adequate for the crews. These relate to the grouping of assignments for a same type of shift within consecutive days (or shifts starting at the same time) or the allocation of compensation days after rest days. For the sake of fairness, the workload should also be balanced over the planning period, both in term of numbers of assignments on the types of shifts and in term of working time differences since not all the types of working shifts have the same duration.

The minimum demands ensure a minimum overall capacity over the forest fire period. However, if there exists a room for improvement while respecting all of the strict constraints expressed, the institution wishes to balance the extra assignments over the types of shift. For a day, it would be preferable to balance the operational capacity over the types of shift rather than assigning all the crews that can be mobilized above the minimum demands on a unique type of shift.

To the best of our knowledge, the timetabling problem of firefighter crews for institutions whose mission is to fight against forest fires has not been yet studied in the literature. Due to climate change, the forest fire period increases and forest fire grows in number and intensity. Increasing the operational capacity of firefighter crews is likely to become an issue. One way to do this is to build better crew plannings by taking into account work regulation constraints but also good practice constraints to ensure fairness. The contributions of this work are summarized as follows:

- first, we propose an Integer Linear Programming model for the **F**ire**F**ighters **T**imetabling **P**roblem (FFTP) that we address. The ILP is designed for modeling purposes and with the aim of obtaining good quality solutions for some instances to be used as reference solutions. However, ILP solvers may face difficulties in attaining feasible solutions within reasonable processing times;
- second, we use the ILP as a basis to put forward a matheuristic. We propose three neighborhoods to make the solver working on subproblems with the aim of obtaining feasible solutions for all of the instances within reasonable processing time. We propose to explore the search space using a variable neighborhood descent (VND) approach based on the three neighborhoods;
- third, we propose an Adaptive Large Neighbourhood Search metaheuristic (ALNS) to investigate a second solution approach. We conduct preliminary experiments to tune the parameters of the components of the ALNS. We make use of a version of the matheuristic as one of the construction method. We investigate the contribution of components of the ALNS.
- finally, we show that the ALNS solution approach obtains all of the optimal solutions that can be attained by the ILP, and better solutions than those of matheuristic are obtained within shorter processing time. As such, it can serve as a good basis to improve the operational capacity of firefighter institutions over the forest fire period.

The remainder of this paper is organized as follows. In Section 2 an overview of related work on personnel scheduling problems is presented. Section 3 presents the constraints and the parameters of the **F**ire**F**ighters **T**imetabling **P**roblem that we address. Section 4 presents the ILP that we use to provide a formal model, and Section 5 describes the ILPH matheuristic we put forward based on the ILP model. The ALNS that we propose to address the problem we face here is described in Section 6 and several components are explained in detail. The computational experiments are reported and commented in Section 7 and the effectiveness of components of the ALNS is displayed. The conclusion and perspectives for further research direction are to be found in Section 8.

2 Related work

In a wide range of situations like health care, protection services, railways or warehouse, employees or crews are required to work in different shifts to cover demands. Thus, many different versions of **P**ersonnel **S**cheduling **P**roblems (PSP) have been described in the literature. In the reviews, hundred of papers addressing different kinds of problems are classified (Afshar-Nadjafi (2021); Brucker et al. (2011); Burke et al. (2004); De Bruecker et al. (2015); Ernst et al. (2004a,0); Heil et al. (2020); Qin et al. (2015); Van den Bergh et al. (2013)).

Usually, in the literature, cyclic cases are distinguished from acyclic cases. For example, in the cyclical case, a one-week schedule must be set initially and identically repeated over the planning horizon. In the acyclic case, the planning must be directly constructed by considering all the planning horizon. The

choice of considering a cyclic or acyclic schedule depends on the specifications of the problem. The addressed FFTP in this paper is acyclic, since it is all the planning horizon of several months of the forest fires period must be considered. The INFOCA institution prefer to have acyclic schedules in order to have more flexibility and a better schedule considering its specific constraints. For example, firefighter crews preferences are specified for all the days of the planning horizon and cannot be cyclic. Also, compensation days assignments differ over the planning horizon since the firefighter crews must work in order to accumulate compensation days, and this constraint made the addressed problem acyclic.

Some of PSP have received most attention because of publicly available datasets that can be used to compare search methods (Curtois (2014); Fages and Lapègue (2014); Krishnamoorthy et al. (2012); Lapègue et al. (2013); Smet et al. (2014)). For instance, the shift minimization personnel task scheduling problem (SMPTSP) consists in assigning tasks to multi-skilled employees, tasks need to be assigned to shifts that are already predefined with the aim to minimize the total number of assigned employees. Equity objective and the scheduling of breaks may also be considered as for the Shift Design and Personnel Task Scheduling Problem with Equity objective (SDPTSP-E) defined in Lapègue et al. (2013).

Table 1 presents a synthetic overview of some recent works on acyclic problems.

Table 1 Related studies on acyclic personnel/crew scheduling problems.

No.	Literature	Problem characteristics							Optimization method
		Skill	Shift	Task	P/C	H	Data	Obj	
1	Guerriero and Guido (2022)	✓	✓		P	1w	RD	MO/SO	MO EA: MIP
2	Zucchi et al. (2021)	✓	✓		P	2w	RD	SO	EA: MIP
3	Chandrasekharan et al. (2021)	✓	✓	✓	P	1d	LB	SO	MA
4	Kletzander and Musliu (2020)	✓	✓	✓	P	2-52w	LB	SO	MH: SA
5	Porto et al. (2019)	✓	✓		P	1w	RD	SO	EA: MIP
6	Tadumadze et al. (2019)			✓	P	1d	RD	SO	EA: MIP H
7	Hoffmann and Buscher (2019)		✓	✓	C	1d	RD	SO	EA: ILP
8	Demirović et al. (2019)		✓		P	1-52w	LB	SO	MO MaxSat
9	Hojati (2018)	✓	✓	✓	P	1d	LB	SO	H:G
10	Pour et al. (2018)		✓	✓	C	10-40d	RD	SO	EA: CP MIP
our			✓		C	135d	RD	SO	EA: ILP MA MH: ALNS

Note :

- Skill, Shift, Task, P/C(Person, Crew), H(Horizon, **w**eek/**d**ay).
- Data: RD(Real Data), LB(Literature Benchmark).
- Obj: SO(Single-Objective), MO(Multi-Objective).
- Optimization method: EA(Exact Algorithm), H(Heuristics), MA(Matheuristic), MH(Metaheuristics), MO(Model).
- G(Greedy), CP(Constraint Programming), ILP(Integer Linear Programming), MIP(Mixed Integer Program), ALNS(Adaptive Large Neighborhood Search).

Guerriero and Guido (2022) proposed models to address staff scheduling problems by taking into consideration new criteria involved by the Covid-19 pandemic situation. The initial model aims at optimizing on-site and remotely work days, considering constraints as limitation of office capacities. The authors next proposed derived MIP models to investigate scenarios. To optimize the criteria associated with a scenario, the implemented MIP model

makes use of a sum of weighted terms. Computational results are obtained using real data of a department of the University of Calabria (Italy).

Zucchi et al. (2021) also investigate staff scheduling problems during Covid-19 pandemic for a company that provides pharmaceutical products to hospitals. The objective is to minimize the sum of the deviations from the contractual amount of working hours of each worker. The authors proposed a MIP formulation with constraints that aim at limiting a contagion risk which is estimated using a network of relationships between employees.

Chandrasekharan et al. (2021) proposed a constructive matheuristic (CMH) to address SMPTSP based on decomposition where sub-problems are solved to optimality using integer programming. An employee-based decomposition and a time-based decomposition are proposed. Based on this later, an automatic time-based constructive matheuristic is derived. The automatic time-based CMH proved to be efficient, high quality solutions over all datasets are obtained and for some instances new optimal solutions are attained.

Kletzander and Musliu (2020) proposed a framework to solve the general employee scheduling problem using simulated annealing (SA). Different problems from literature that cover different types of demand and constraints are investigated. All the hard constraint violations are penalized by using a specific hard constraint weight provider which has been used to tune the weights for each data set. The authors implemented a set of moves in the general framework. The approach obtained good results against tailored algorithms.

Porto et al. (2019) evaluated the potential benefits of incorporating labor flexibility into personnel scheduling in a retail store context. The objective is to minimize the levels of over and understaffing considering a multiskilled workforce that has flexible contract to achieve cost-effective operations. The authors proposed a MIP model and investigated scenarios for evaluating several strategies on human resource management.

Tadumadze et al. (2019) investigated mixed-integer models for integrating workforce planning and truck scheduling. The problem is similar to multi-mode RCPSP with additional constraints involved by a truck distribution center (e.g. cross-docking), moreover truck-related objectives are specific. The authors also proposed a formulation based on interval scheduling, next investigated heuristics for interval selection. Three alternative workforce planning approaches for different settings and scenarios are compared.

Hoffmann and Buscher (2019) studied a railway crew scheduling problem with attendance rates. The aim is to find a minimum cost shift schedule satisfying operating conditions and taking into account work regulation constraints. The authors proposed an arc flow ILP model and valid inequalities. Small-sized instances are solved to optimally, valid inequalities proved to be effective to speed-up the solution process and improved the bounds.

Demirović et al. (2019) investigated modeling based on weighted partial boolean maximum satisfiability (maxSAT) problems for the PSP variants introduced by Curtois (2014). A comparison of different cardinality constraint encodings is performed to show their applicability. Optimal solutions are

obtained for two of the instances and obtained solutions for two very large instances. The results for many instances are currently not competitive when compared to tailored solution approaches based on ILP.

Hojati (2018) proposed a greedy heuristic solution method for the SMPTSP. A reduced problem is solved iteratively by selecting the best possible assignment of tasks to one worker. At each iteration, the worker with maximum objective value is chosen. For very large instances, the proposed Greedy heuristic method performs very well compared to other solution approaches that require a commercial ILP Solver.

Pour et al. (2018) addressed a preventive signal maintenance crew scheduling problem (Danish railway). The authors proposed a MIP model, a Constraint Optimization Problem (COP) model, and a hybrid CP/MIP solution approach. The hybridization of CP to build an initial solution for warm-starting the MIP obtained the best results.

After presenting recent works on PSP addressed in the literature, we move on to discussing the techniques employed for their solution. Table 1 shows that many different search methods have been applied, ranging from exact methods, matheuristics, heuristics, metaheuristics, and hybrid techniques. Exact methods are able to find optimal solution for small and medium size instances but may face difficulty in obtaining solution for larger instances within reasonable processing time. Matheuristics can help with this problem according to the characteristics of the problem. However, in some cases, heuristics or metaheuristics approaches are need to obtain solutions of good quality for larger instance.

There are a large variety of PSP, ranging from well defined problems, on the basis on real problems, for which data sets are publicly available, to specific ones that are addressed to respond to a need. The problem we are dealing with correspond to this latter case.

In our problem, a crew can be viewed as a super employee composed of people with multiple highly specialized skills. These people constitute an extremely welded unit, trained to have group automations to be effective in extreme situations. The number of crew is fixed and not to be minimized. On the contrary, the purposes are to increase operational capacity while considering a limited number of crew, to maintain equity between crews and to make possible the grouping of day-off when feasible.

3 Problem definition

Firefighters do dangerous and difficult work, the institution aims to increase operational capacity but at the same time wants to respect as much as possible the preferences they express and all of the soft constraints that aim to build a fair timetable.

A formal definition of the addressed FFTP problem can be set as follows: we consider a set of crews C, a set of shifts S, a set of days D, a set of hard constraints HD and a set of soft constraints SC. The problem consists in

producing an assignation for each crew $c \in C$ to a shift $s \in S$ for each day $d \in D$ while satisfying all the hard constraints $hd \in HD$ and minimizing the soft constraints violations $sc \in SC$.

We give the set of daily working shifts to be considered, we introduce the hard constraints to be respected and the soft constraints used to assess the quality of a solution. The notation used for the types of shifts are as follows:

(T12) from 8 am to 4 pm at fire station, regular daily shift;
(T16) from 3 pm to 10 pm at fire station, regular daily shift;
(H) from 8 am to 4 pm at fire station, regular daily shift, assigned to a helicopter;
(N) from 10 pm to 8 am at fire station, regular night shift;
(G7) from 7 am to 3 pm at fire station, stand-by to face instantly any extra urgent request;
(G24) 24h guard, crew stay at home but may be mobilized to face any urgent situation;
(A3) from 8 am to 6 pm at fire station (or elsewhere) for training purposes;
(R) rest day;
(C) additional compensation day granted when a number of hours have been worked.

The hard constraints that relate to work regulation and to local regulation of the INFOCA institution are as follows:

(H1) **one shift a day:** a firefighter crew can only be assigned to one shift a day;
(H2) **minimum demands:** each daily shift has a minimum demand of firefighter crews;
(H3) **forbidden shift successions:** some shift assignments on consecutive days are forbidden;
(H4) **maximum workload:** over the planning horizon, a maximum workload for every crew should not be exceeded;
(H5) **compensation:** compensation days are granted according to the hours worked, they should be used;
(H6) **maximum consecutive working days:** every firefighter crew has a maximum number of consecutive working days.

Some consecutive shift assignments are forbidden for a crew, for instance a night shift (N) ends at 8 am and cannot be followed by any shift which begins at 8 am.

Figure 1 shows a representative example with six crews (C1 to C6), three types of shift and a minimum demand of one for each type of shift. The types of shift are T12 (vertical hatch), T16 (diagonal hatch) and N (black). The forbidden shift successions are (N, T12) and (N, T16). Figure 1 depicts a part of the timetable computed using the $BuildFeasibleSchedule()$ heuristic we present in Section 6.

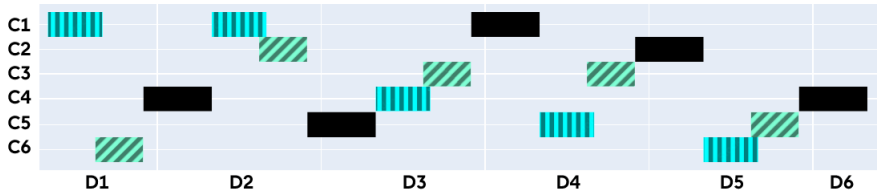

Fig. 1 A representative example with six crews and three types of shift.

Soft constraints are constraints of good practice that should be satisfied as best as possible. The violation of any soft constraint induces a penalty. A weighted sum of the penalties measures the quality of the solution produced. For the studied firefighters timetabling problem, the soft constraints are the following:

(S1) **shift grouping:** assignments of a crew to the same shift should be grouped. Each shift assignment change between two consecutive days is penalized;
(S2) **same start time:** start times should be the same whatever the working shifts over consecutive working days. Each starting time change for working shifts between two consecutive days is penalized;
(S3) **compensation assignments:** compensation day assignments should be right after the rest days, the aim is to allow firefighters to have a short vacation during the planning period. Each assignment of compensation not right after rest days is penalized.
(S4) **period fairness:** for the sake of fairness the workload should be balanced between the crews over the planning period. The unbalance of workload between crews should be minimized;
(S5) **preferences:** each crew assignment to an undesired shift is penalized;
(S6) **evenly balance extra daily shifts:** assigning of extra crews to the different types of shifts should be balanced each day. The unbalance on extra assignment to different shifts should be minimized each day.

The increase of overall operational capacity may not be obtained at the expense of what is considered as a good timetable by the crews. The shift grouping (S1) and the same start time (S2) are soft constraints that aim at produce a more regular timetable for a crew.

The compensation (H5) hard constraint and the maximum consecutive working days (H6) hard constraint are institutional one's, the later enforces the rest days (R). Since compensation days (C) are to be granted, the compensation assignment (S3) soft constraints aims at grouping the compensation days right after the rest days, that makes it possible to obtain larger off-periods.

The period fairness (S4) soft constraint relates to equity both in the amount of hours worked and to equity in the number of working shifts that are assigned. These concerns have to be faced by the institution to be equally fair with all of the crews.

Although they are trained to perform any shift, some crews can express preferences, this is the soft constraint preference (S5). This is important for the institution to take (S5) into consideration.

Provided the minimum demand (H2) is respected, the idea beyond (S6) is to ensure a balance between shift assignments. If we can assign three extra crews in a day, it is preferable to assign the crews to three different types of shifts rather than assigning the three crews to a same type of shift.

This makes it possible to refine the quality of the solution by better spread crews over the types of shift.

The main goal is the increase of overall operational capacity over the forest fire period while providing to crews fair and good timetables.

4 ILP model for FFTP

In this section we present the ILP model for minimizing the criteria we detailed in Section 3. We first introduce data and parameters that correspond to the description of the problem, then the main decision variables that we use to check the hard constraints (H1)-(H6). Next, we introduce data, parameters and variables we use to assess the soft constraints.

Data and parameters:

$\mathcal{C}$ set of firefighter crews, size n_c;
$\mathcal{D}$ set of days of the planning period, a day $d \in [1, \cdots, n_d]$, size n_d;
$\mathcal{D}^-$ set of days of the planning period, except the last day n_d;
$\mathcal{S}$ set of types of shifts, $\{T12,\ T16,\ H,\ N,\ G7,\ G24,\ A3, R,\ C\}$, size n_s;
$\mathcal{S}_w$ set of types of working shifts, $\{T12,\ T16,\ H,\ N,\ G7,\ G24,\ A3\}$, size n_w;
F set of couples of forbidden consecutive shift assignments, e.g. $(N, H) \in F$;
r_s daily minimum demand for a working shift $s \in \mathcal{S}_w$;
l_s duration of shift s (length in hours);
L maximum workload for any crew over the planning period;
MAX_d maximum number of consecutive work days for a crew (see H6);
WHC number of worked hours giving a compensation day.

The primary boolean and integer decision variables are as follows:

- $X_{csd} \in \{0, 1\}$, one if crew c works on shift s in day d, zero otherwise;
- $\rho_{cd} \in \mathbb{N}$ number of worked hours of crew c from the first day to day d.

Compensation days are granted using ρ_{cd} integer variables.

Data and parameters that relate to soft constraints are as follows:

t_s start time of shift s;
$p_{csd} = 1$ if crew c does not prefer to work on shift s on day d, zero otherwise.

The main objective is to increase overall operational capacity while minimizing the soft constraints violations. We introduce first the variables we used to assess the operational capacity and those we use to assess the (S1)-(S3) soft constraints violations:

- $\lambda_d \in \{0, \cdots, n_c\}$, for a day d, the difference between the maximum number of assignable crew (n_c) to working shifts $\mathcal{S}_w$ and those assigned;
- $\alpha_{cd} \in \{0, 1\}$, one if crew c works on shift s in day d and works on a different shift s' in day $d+1$, zero otherwise (see S1, shift grouping);
- $\beta_{cd} \in \{0, 1\}$, one if crew c works on shift s in day d and works on a different shift s' in day $d+1$ with $t_s \neq t_{s'}$, zero otherwise (see S2, same start time) ;
- $\gamma_{cd} \in \{0, 1\}$, one if the crew c works on shift s in day d with $s \neq R$ and is assigned to shift $s' = C$ in day $d+1$, zero otherwise (see S3, compensation assignment).

At most n_c crews per day can be assigned to working shifts $\mathcal{S}_w$. Hence, minimizing the sum of the λ_d is the same as maximizing the overall operational capacity all over the horizon.

The equity concern relates to fairness in the amount of hours worked and to fairness in the number of working shifts that are assigned (see S4, period fairness). For a crew c, we introduce integer variables as follows:

- $\theta_c \in \mathbb{N}$, total working time of crew c over the planning period;
- $\delta_c \in \mathbb{N}$, total number of worked shifts for crew c over the planning period.

We introduce the last integer variables as follows:

- $\phi_{cc'} \in \mathbb{N}$, number of shift assignment difference between the crews c and c';
- $\varphi_{cc'} \in \mathbb{N}$, working time difference between the crews c and c';
- $\psi_{ss'} \in \mathbb{N}$, unbalance of assignments between the s and s' shifts.

The first two variables are used to assess the two concerns on fairness. The $\psi_{ss'}$ variables are used to address "evenly balance extra daily shifts" (S6). No additional variables are required to assess the crews' preferences (S5).

Finally, we introduce the weights as follows:

- w_{oc} **o**perating **c**apacity weight;
- w_{sg} **s**hift **g**rouping violation weight (S1);
- w_{sst} **s**ame **s**tart **t**ime change violation weight (S2);
- w_{ca} **c**ompensation **a**ssignments violation weight (S3);
- w_p **p**references violation weight (S5).

We propose the following ILP to address this problem:
Min

$$w_{oc} \cdot \sum_{d \in \mathcal{D}} \lambda_d \tag{1a}$$

$$+ \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} (w_{sg} \cdot \alpha_{cd} + w_{sst} \cdot \beta_{cd} + w_{ca} \cdot \gamma_{cd}) \tag{1b}$$

$$+ w_p \cdot \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}_w} \sum_{d \in \mathcal{D}} p_{csd} \cdot X_{csd} \tag{1c}$$

$$+ \sum_{c \in \mathcal{C}} \sum_{c' \in \mathcal{C}} (\phi_{cc'} + \varphi_{cc'}) \tag{1d}$$

$$+ \sum_{s \in \mathcal{S}_w} \sum_{s' \in \mathcal{S}_w} \psi_{ss'} \tag{1e}$$

Subject to:

$$X_{csd},\ \alpha_{cd},\ \beta_{cd},\ \gamma_{cd} \in \{0,1\} \tag{2}$$

$$\lambda_d, \phi_{cc'}, \varphi_{cc'}, \psi_{ss'}, \delta_c, \theta_c, \rho_{cd} \in \mathbb{N} \tag{3}$$

$$\sum_{s \in \mathcal{S}} X_{csd} = 1 \quad \forall c \in \mathcal{C},\ \forall d \in \mathcal{D} \tag{4}$$

$$\sum_{c \in \mathcal{C}} X_{csd} \geq r_s \quad \forall d \in \mathcal{D},\ \forall s \in \mathcal{S}_w \tag{5}$$

$$X_{csd} + X_{cs'(d+1)} \leq 1 \quad \forall (s,s') \in F,\ \forall c \in \mathcal{C},\ \forall d \in \mathcal{D}^- \tag{6}$$

$$\sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} l_s \cdot X_{csd} \leq L \quad \forall c \in \mathcal{C} \tag{7}$$

$$\sum_{s \in \mathcal{S}_w} \sum_{d' \in \mathcal{D}, d' \leq d} l_s \cdot X_{csd} = \rho_{cd} \quad \forall c \in \mathcal{C},\ \forall d \in \mathcal{D} \tag{8}$$

$$WHC \cdot \sum_{d' \in \mathcal{D}, d' \leq d} X_{csd} \leq \rho_{cd} \quad s = \mathrm{C},\ \forall c \in \mathcal{C},\ \forall d \in \mathcal{D} \tag{9}$$

$$\sum_{d \in \mathcal{D}} X_{csd} = \left\lfloor \frac{\rho_{c(l_d)}}{WHC} \right\rfloor + 1 \quad s = \mathrm{C}, \forall c \in \mathcal{C} \tag{10}$$

$$\sum_{s \in \mathcal{S}_w} \sum_{d' \leq 1 + MAX_d,\ d + d' \leq n_d} X_{csd} \leq MAX_d \quad \forall c \in \mathcal{C},\ \forall d \in \mathcal{D} \tag{11}$$

$$\sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}_w} X_{csd} = n_c - \lambda_d \quad \forall d \in \mathcal{D} \tag{12}$$

$$X_{csd} + X_{cs'd+1} \leq 1 + \alpha_{cd} \quad \forall c \in \mathcal{C}, \forall d \in \mathcal{D}^-, \forall s, s' \in \mathcal{S}_w, s \neq s' \tag{13}$$

$$X_{csd} + X_{cs'd+1} \leq 1 + \beta_{cd} \quad \forall c \in \mathcal{C}, \forall d \in \mathcal{D}^-, \forall s, s' \in \mathcal{S}_w, s \neq s' \text{ with } t_s \neq t_{s'} \tag{14}$$

$$X_{csd} + X_{cs'd+1} \leq 1 + \gamma_{cd} \quad \forall c \in \mathcal{C},\ \forall d \in \mathcal{D}^-\ \forall s \in \mathcal{S}_w, s' = \mathrm{C} \tag{15}$$

$$\sum_{s \in \mathcal{S}_w} \sum_{d \in \mathcal{D}} X_{csd} = \delta_c \quad \forall c \in \mathcal{C} \tag{16}$$

$$\sum_{s \in \mathcal{S}_w} \sum_{d \in \mathcal{D}} l_s \cdot X_{csd} = \theta_c \quad \forall c \in \mathcal{C} \tag{17}$$

$$\delta_c - \delta_{c'} \leq \phi_{cc'} \quad \forall c, c' \in \mathcal{C},\ c \neq c' \tag{18}$$

$$\theta_c - \theta_{c'} \leq \varphi_{cc'} \quad \forall c, c' \in \mathcal{C},\ c \neq c' \tag{19}$$

$$\left(\sum_{c \in \mathcal{C}} X_{csd} - r_s\right) - \left(\sum_{c \in \mathcal{C}} X_{cs'd} - r_{s'}\right) \leq \psi_{ss'} \quad \forall d \in \mathcal{D},\ \forall s, s' \in \mathcal{S}_w \tag{20}$$

The five terms of the objective function aim at maximizing operational capacity while minimizing the soft constraints violations. The terms (1a) to (1e) refers to the penalization of the soft constraints presented in Section 3.

The hard constraints (H1) to (H6) are enforced by Constraints (4)-(11). The Soft constraints (S1) to (S6) are enforced by Constraints (12)-(20).

This model is an improvement of a first model presented in Ouberkouk et al. (2021), fewer variables are used and we achieve better results on the smaller instances. For the ILP presented in Ouberkouk et al. (2021), the number of each variable α, β and γ is equal to $n_c \cdot n_s \cdot n_s \cdot n_d$. Here, the number of each of these variables is equal to $n_c \cdot n_d$.

In the following, we denote $Obj(S)$ as the function that computes the quality of a solution S as presented in equations (1a)-(1e).

5 ILP based matheuristic

ILP solvers may face difficulties to run the proposed ILP model as the instances grow in size, and feasible solutions may be difficult to obtain within reasonable processing times. The use of ILP solvers in a heuristic context for producing good solutions has been investigated for some personnel scheduling problems as nurse rostering (Santos et al., 2016), maintenance crew scheduling (Pour et al., 2018) and aircraft maintenance (De Bruecker et al., 2018). The general idea is to make the solvers working on subproblems so as to perform local searches with the aim of obtaining good feasible solutions within shorter processing time than the ILP does (fix-and-optimize strategy).

The ILP based matheuristic that we propose consists of two subsequent phases: a construction phase to create a subproblem and a local search phase that uses the ILP we proposed in Section 4.

Given a feasible current solution, a subproblem is obtained by fixing a subset of variables that relate to firefighter crews allocations (hard-fixation), next the subproblem is solved to optimality unless a given time limit is attained. A subproblem allows us to explore a certain neighborhood of a solution and we propose to investigate three neighborhoods: Fix Day, Fix Shift and Fix Crews.

The neighborhoods we propose create subproblems by fixing of a given set of variables of a current solution. Next, the ILP based matheuristic, we denoted as ILPH, is presented. The overall idea is to explore the search space

using a variable neighborhood descent (VND) approach based on these three neighborhoods.

Fix Day neighborhood

The subproblems are generated by fixing all the firefighter crews allocations on $|\mathcal{D}| - Size_w$ days in the planning period where $Size_w$ is the size of the sliding window. Thus, the ILP can be used to optimize the allocations over $Size_w$ days which is a parameter of the neighborhood.

At the first iteration ($i = 0$), a subproblem is created from an initial solution by fixing all crews' allocations but the ones on the days from $day_A = 1$ to $day_B = Size_w$. The subproblem is then solved using the ILP and a new solution is obtained.

The sliding of the $Size_w$ window is managed by updating the beginning day_A and the end day_B of the window as follows:

$$day_A = 1 + i \,.\, Slide_w \qquad\qquad day_B = Size_w + i \,.\, Slide_w$$

where i is the number of the current iteration and $Slide_w$ is the second parameter that defines by how many days the window slides between two consecutive subproblems. When either of day_A or day_B is larger than n_d, the value is set at l_d.

The idea is to slide the window and solve to optimality each subproblem in turn, whether feasible. The value of parameter $Slide_w$ is used to determine $n_d/Slide_w$ the number of subproblems to be solved that is also the number of iterations. We compute an incumbent solution S_i at iteration i, the best solution found so far S_{best} can be then updated when the solution S_i is an improvement.

The smaller the value of $Slide_w$, the greater the number of subproblems explored in the neighborhood is. The size of each subproblem is determined by $Size_w$. Small values may create subproblems that do not contain any better solution. Large values may create subproblems for which the solver may face difficulty in attaining optimal solution within short processing time. Inside the window, where all the crews' allocations are free, all the terms of the objective function are equally considered and none is favored.

Fix Shift neighborhood

The subproblems are created by fixing all the allocations of crews that relate to all types of shifts but one. The crews' allocations that relate to a type of shift are free. This tends to improve for instance the shift grouping (S1) and to evenly balance of extra daily shifts (S6). We first sort the type of shifts by their decreasing penalty values in the current solution. Given a sorted list, at the first iteration, only the crews' allocations that relate to the first shift type are free, and so on. There are n_s iterations since the number of different subproblems generated in this neighborhood is equal to the number of types of shifts.

Given an incumbent solution S_i, it is improved, if any. The best solution found so far S_{best} can be then updated.

Fix Crew neighborhood

The subproblems are created by fixing all the allocations of crews that relate to all the crews but one. The crews' allocations that relate to a crew are free. This tends for instance to improve the compensation assignments (S3) (See term (1b)). We sort first the crews by their decreasing penalty value in the current solution. Given a sorted list, at the first iteration, only the crews' allocations that relate to the first crew are free, and so on. There are n_c iterations since the number of different subproblems generated in this neighborhood is equal to the number of crews.

Given an incumbent solution S_i, it is improved, if any. The best solution found so far S_{best} can be then updated.

Mathematical Integer Programming Heuristic

We conducted preliminary experiments on the neighborhoods one at a time. We practically observed that the Fix Day neighborhood may obtain more gains than the Fix Shift and Fix Crew neighborhoods, but at the expense of larger processing time. This was expected for the Fix Day neighborhood because as the window size increases the number of unfixed variables increases. The two later neighborhoods focus on what can be improved by considering one by one either the crews or the shifts, more variables are fixed, which limits the possibilities of gain but the processing times are shorter. We have therefore decided to give priority to Fix Day neighborhood and when an improvement is obtained to seek additional gains with the other two.

Algorithm 1 shows the proposed Mathematical Integer Programming Heuristic we denote as ILPH.

Given an initial solution S_{cur}, the Fix Day neighborhood is explored by performing the while loop that increases the sliding window. When an improvement is met the two other neighborhoods apply. We limit the processing time of runs using T_l. The time given for a run of $Explore\_Fixday\_Neighborhood$ on a window is twice the time for the others because more gains are expected (see line 7). Since the $Explore\_Fixshift\_Neighborhood$ (see line 10) sorts the shifts in the decreasing order of potential gains, at most n_s number of runs of the ILP may be performed. That makes it possible to obtain the most important gains before T_l is attained. The same rationale applies for $Explore\_Fixcrews\_Neighborhood$ (see line 12).

The Fix Day neighborhood does not favor any term of the objective function while Fix Shift neighborhood is a "type of shift" oriented and Fix Crew neighborhood is a "crew" oriented.

Algorithm 1 first explore small windows, next we explore larger ones. Our purpose is to gradually improve the solution so that we obtain solution evaluations that will help in pruning the search as the windows increase. This also makes windows overlapping over the iterations of the ILPH possible. For the

Input : S_{cur} initial solution, T_{ILPH} global time limit, T_l time limit used to stop the $Explore\_*$
Output : S_{best} best solution found
Variables : S_i solution at iteration i, $Size_w$ size of the sliding window, $Slide_w$ by how many days the window slides

$i := 0$
$S_{best} := S_{cur}$
/* S_i used to fix values of some variables prior optimize */
$S_i := S_{cur}$
$Slide_w := 2$
while *($Slide_w \leq n_d/2$) and (T_{ILPH} not reached)* **do**
  $Size_w := 2 \,.\, Slide_w$
  /* $Size_w$, $Slide_w$ and i, used to fix variables */
  /* At iteration i, the run stops when $2 \cdot T_l$ is attained */
  $S_i := Explore\_Fixday\_Neighborhood(Size_w, Slide_w, i, S_i, 2 \cdot T_l)$
  if $Obj(S_i) < Obj(S_{best})$ **then**
    $S_{best} := S_i$
    /* At most n_s runs, stop when T_l is attained */
    /* All shift assignments are fixed except one */
    $S_i := Explore\_Fixshift\_Neighborhood(S_i, T_l)$
    if $Obj(S_i) < Obj(S_{best})$ **then** $S_{best} := S_i$
    /* At most n_c runs, stop when T_l is attained */
    /* All crew assignements are fixed except one */
    $S_i := Explore\_Fixcrews\_Neighborhood(S_i, T_l)$
    if $Obj(S_i) < Obj(S_{best})$ **then** $S_{best} := S_i$
  end
  $Slide_w := Slide_w + 1$
end
return S_{best}

Algorithm 1: ILPH

sake of simplicity, we choose to link the two parameters as $Size_w = 2 \,.\, Slide_w$ and we set the initial value of $Slide_w$ to 2. The loop increases the value of $Slide_w$ by 1. Provided that a global time limit T_{ILPH} is attained or that the last $Slide_w$ value is larger than $n_d/2$, the ILPH stops.

The ILPH that we propose will be evaluated as a matheuristic to deal with the FFTP. In addition, some components of the ILPH can also be viewed as a neighborhood component that can also be embedded within the ALNS approach that we propose further, this will also be investigated.

6 ALNS metaheuristic for the FFTP

Solvers may faced difficulties in attaining good enough solutions within short processing times running the ILP model and the ILPH matheuristic.

Metaheuristic based solution approaches have been extensively reported in literature to address a large variety of optimization problems, for a comprehensive survey we invite the reader to refer to Hussain et al. (2019). We propose an Adaptive Large Neighborhood Search meta-heuristic (ALNS) to compute solutions of good quality for the FireFighters Timetabling Problem (FFTP).

For a more general and recent review on the ALNS metaheuristic framework and its applications, we invite the reader to refer to Mara et al. (2022).

We present in Algorithm 2 its general structure prior to provide insights on the components.

Input : An instance of FFTP
Output : S_{best} best solution found
Parameters: D_{limit} limit for diversification degree,
Variables : $MaxIter$ maximum number of iterations without any improvement prior stopping the ALNS
$AcceptIter$ number of iterations without any improvement prior accepting a degradation,
D_{max} current diversification degree, M_d, M_c destruction and construction methods,
S_{old} current solution to be improved, S_{cur} solution under work

$i, i' := 0$ `/* i, i' number of iterations */`
$MaxIter := \varepsilon.n_c$
$AcceptIter := n_c$
$D_{max} := 3$
S_{best}, S_{old} := BuildFeasibleSchedule()
while $i < MaxIter$ **do**
  $M_d := ChooseDestructionMethod()$
  $k :=$ rand(1,D_{max})
  $S_{cur} := AdaptiveDestruction(S_{old}, k, M_d)$
  $M_c := ChooseConstructionMethod()$
  $S_{cur} := ApplyConstruction(S_{cur}, M_c)$ `/* insert as many crews as possible in` S_{cur} `*/`
  if $Obj(S_{cur}) < Obj(S_{best})$ **then**
    S_{best}, $S_{old} := S_{cur}$
    $i, i' := 0$
    $D_{max} := 3$
  else
    if $(i' \geq AcceptIter$ *and* $AcceptDegradation(S_{cur}, S_{old}))$ **then**
      $S_{old} := S_{cur}$
      $i' := 0$
    end
    $i++$
    $i'++$
    $D_{max} :=$ Min(D_{max}+1,D_{limit})
  end
  $UpdateDestructionsScores()$
end

Algorithm 2: General structure of ALNS for FFTP

An initial solution S_{old} is computed using the greedy constructive heuristic $BuildFeasibleSchedule()$. The objective of $BuildFeasibleSchedule()$ is to obtain a feasible solution without taking into account the objective function. The general idea is to take crews in turn at random, next each crew is assigned to the working shifts while complying with the hard constraints (H1 and H3-H6). The algorithm stops as soon as the **minimum demands** (H2) is respected. This greedy heuristic is repeated until a feasible solution is met. We

obtain a feasible initial solution but there is room for improvement especially in operational capacity.

A destruction method M_d is selected at random from among **R**andom **D**estruction and **S**mart **D**estruction using $ChooseDestructionMethod()$. The **S**mart **D**estruction method makes use of the **B**est **I**nsertion **D**estruction **C**riterion (BIDC) to assess the impact of unassigning of crews.

We remove $k \leq D_{max}$ crews at each iteration. We define D_{max} as the degree of diversification. This value is initialized to three and then incremented after each non-improving iteration up to D_{limit}. We choose to set $D_{limit} = \lceil n_c/n_s \rceil$ that represents the average number of crews that can be assigned to a type of shift for a day. Provided an improvement is found, we reset D_{max} to three to entirely explore the neighborhood of the new solution. This adaptive diversification mechanism makes it possible to broaden the search around a solution with the aim to obtain a better solution.

We obtain the current solution under work S_{cur} by applying $AdaptiveDestruction(S_{old}, k, M_d)$. Some crews have been unassigned, and some others may be not assigned yet. We therefore can use these crews to find a better solution.

A construction method M_c is selected at random by using $ChooseConstructionMethod()$. The construction methods aim at completing and improving the current solution. We implement an adaptive **B**est **I**nsertion **A**lgorithm (BIA) that also assess feasible assignments using the **B**est **I**nsertion **D**estruction **C**riterion (BIDC). We also use as a construction method a version of the matheuristic ILPH (see Section 5). We then insert as many crews as possible while respecting the hard constraints using $ApplyConstruction(S_{cur}, M_c)$.

To avoid to be stuck in local optima the ALNS approach requires an acceptance procedure that makes feasible to select a solution of low quality with the aim to explore other parts of the search space. We use a record-to-record-based approach (Dueck, 1993). Provided that a number $AcceptIter$ iterations without any improvement is met, and provided that a lower quality solution is accepted by $AcceptDegradation(S_{cur}, S_{old})$, we make feasible to continue exploring the search space using this lower quality solution. We choose to set $AcceptIter$ to n_c, the general idea is to increase the processing time as n_c the number of crew increases.

The success of destruction methods may vary depending of the instance. An adaptive choice generally leads to better results rather than fixing the choices once and for all. Each destruction method has a score which represents its share in a wheel. The $UpdateDestructionsScores()$ procedure updates the scores.

When a maximum number of iterations $MaxIter$ without any improvement is met, the ALNS algorithm stops, and then returns the best solution found so far within iterations S_{best}. We choose to set $MaxIter$ to $\varepsilon.n_c$, where ε needs to be tuned to obtain a good trade-off between good quality solutions and processing times.

Adaptive mechanisms for destruction, construction and acceptance procedures are used. Several parameters need to be tuned to obtain a good efficiency of these adaptive mechanisms.

Best Insertion Destruction Criterion

We make use of the **B**est **I**nsertion **D**estruction **C**riterion (BIDC) in the **S**mart **D**estruction method and in the **B**est **I**nsertion **A**lgorithm (BIA). Let be (d, s, c) a triplet for a **d**ay, a **s**hift and a **c**rew. Given a solution we evaluate an unassignment or an assignment of a triplet by computing the BIDC as follows:

$$(SG^{\alpha}.\ SST^{\beta}.\ CA^{\gamma}.\ PF^{\theta}.\ P^{\omega}.\ EB^{\mu})$$

The BIDC is composed of one term for each soft constraints: SG is for the **S**hift **G**rouping (S1), SST is for the **S**ame **S**tart **T**ime (S2), CA is for the **C**ompensation **A**ssignments (S3), PF is for the **P**eriod **F**airness (S4), P is for the **P**references (S5) and EB is for **E**venly **B**alance extra daily shifts (S6). The BIDC value is set to $+\infty$ when a hard constraint is violated. The values of the parameters $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ are managed by an adaptive strategy that relates to the **B**est **I**nsertion **A**lgorithm (BIA), that makes it possible to adapt the relative importance of the terms during the course of the algorithm.

Destruction methods and adaptive mechanism for choosing a destruction method

The $AdaptiveDestruction(S_{old}, k, M_d)$ is used to unassign some crews from S_{old}, the solution to be improved. Given a value k for the number of crews to be unassigned, one among the two following destruction methods apply:

Random Destruction (RD): crews are selected at random;
Smart Destruction (SD): crews are selected using the BIDC.

For the **S**mart **D**estruction method, we consider all the (d, s, c) triplets of assigned crews, and we assess their BIDC scores. We use the parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ that produced the best solution at the previous iteration of BIA (see **Construction methods**). The scores are next used as shares in a roulette wheel that is used to select the k crews to be unassigned. We aim at selecting the assigned crews with highest BIDC in the hope of assigning these crews to better days and shifts.

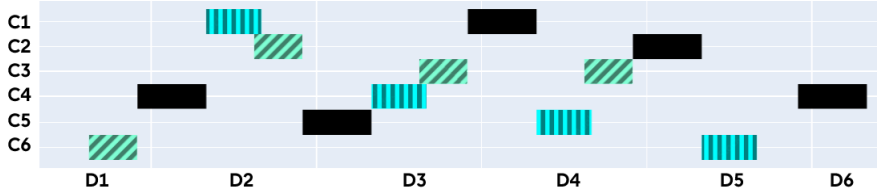

Fig. 2 A destruction method is applied: assignments (D1, T12, C1) and (D5, T16, C5) are removed from timetable depicted in Figure 1.

Figure 2 shows the schedule after applying a destruction method. The minimum demands may not be met temporarily before a construction method is applied. This frees up some crews, that makes it possible to find a better solution.

The selection of a destruction method is managed using an adaptive mechanism. Let $DeSc_{ij}$ be the **De**struction **Sc**ore of method $j \in \{RD, SD\}$ at iteration i. After each iteration i, the destruction scores are updated by $UpdateDestructionsScores()$ as follows:

$DeSc_{ij} := (1+\lambda)DeSc_{(i-1)j}$ if $Obj(S_{cur}) < Obj(S_{best})$;
$DeSc_{ij} := (1+(1/2)\lambda)DeSc_{(i-1)j}$ if $Obj(S_{cur}) < Obj(S_{old})$ and $Obj(S_{cur}) \geq Obj(S_{best})$;
$DeSc_{ij} := (1-(1/2)\lambda)DeSc_{(i-1)j}$ if $Obj(S_{cur}) \geq Obj(S_{old})$ and $Obj(S_{cur}) \geq Obj(S_{best})$;
$DeSc_{ij} := DeSc_{(i-1)j}$ if the destruction method j is not used.

The aim is to favour the destruction method that obtains the best result during the course of the algorithm. However, this cannot be done at the definitive expense of one against the other since the relative effectiveness may depend on the instance and also may change during the course of the algorithm. The parameter λ is used to smooth the strengthening and needs to be tuned to obtain an effective adaptive mechanism.

Construction methods

The $ApplyConstruction(S_{cur}, M_c)$ is used to complete the solution under work S_{cur} by assigning as many crews as feasible. Given a S_{cur}, one of the two following construction methods apply:

ILPH: using only the Fix Day neighborhood;
BIA: using an adaptive mechanism for managing the $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ parameters.

The ILPH matheuristic is time consuming, thus we chose to use only the Fix Day neighborhood to implement a construction method. Moreover, we fixed the time limit T_{ILPH} to one minute so as to avoid to unnecessary waste processing time in ILP solving. The parameter $Size_w$ needs to be tuned to obtain a good trade-off between solution quality and processing time. We recall that the two parameters $Size_w$ and $Slide_w$ are linked as $Size_w = 2\,.\,Slide_w$.

The BIA algorithm (see Algorithm 3) uses S_{cur} the partial solution under work and a parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ as inputs, and it tries to insert as many unassigned crews c as possible. The BIA makes use of the **B**est **I**nsertion **D**estruction **C**riterion (BIDC) to assess all the feasible insertions (d, s, c). The best triplet is retained, if any. The best insertion is performed, then the quality of the solution is assessed. When no more valid insertion is possible, the BIA stops. Given a parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$, a run of BIA returns $S_{curbest}$, the best solution found which can be then possibly used as the new S_{cur} for the next iteration of the ALNS.

Input : S_{cur} a partial solution under work
$(\alpha, \beta, \gamma, \theta, \omega, \mu)$ parameter set
Output : $S_{curbest}$ best solution found over the BIA iterations
Variables : $(\text{d,s,c})^*$ best triplet, Bsuccess boolean

$S_{curbest} := S_{cur}$ `/* store reference solution for BIA */`
Bsuccess := true
while *Bsuccess* **do**
  $(\text{d,s,c})^* := (\emptyset, \emptyset, \emptyset)$
  `/* Using BIDC to assess, find the best triplet, if any */`
  foreach $d \in \mathcal{D}$ **do**
    foreach $s \in \mathcal{S}$ **do**
      foreach $c \in UnassignedCrews(d, s)$ **do**
        ComputeBIDC(d,s,c)
        UpdateBestTriplet $(\text{d,s,c})^*$
      end
    end
  end
  Bsuccess := Insert(S_{cur}, $(\text{d,s,c})^*$) `/* if no feasible insertion return false */`
  `/* Comparing` S_{cur} `and` $S_{curbest}$`, all terms of the objective function are assessed */`
  if $Obj(S_{cur}) < Obj(S_{curbest})$ **then**
    $S_{curbest} := S_{cur}$
  end
end

Algorithm 3: Best Insertion Algorithm

Figure 3 shows the timetable obtained after applying a construction method. The solution complies with the minimum demand hard constraint and more shifts are assigned to crews.

We make use of an adaptive mechanism for managing the construction of a new S_{cur} using the BIA. We run separately four BIA with different values of the parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$. Let be α_{i-1}, β_{i-1}, γ_{i-1}, θ_{i-1}, ω_{i-1} and μ_{i-1} the best parameter values obtained at a previous iteration of the ALNS. The α, β, γ, θ, ω and μ values are chosen randomly in the six dimension space having the center $(\alpha_{i-1}, \beta_{i-1}, \gamma_{i-1}, \theta_{i-1}, \omega_{i-1}, \mu_{i-1})$ and the side length ϕ. The best solution $S_{curbest}$ obtained among the four runs is kept, and the parameter set that produces it is stored to be used for the next iteration. This best parameter set is used by **S**mart **D**estruction when chosen, and is used by BIA when chosen. This adaptive mechanism makes it possible to speed-up the convergence of the ALNS algorithm towards a good solution.

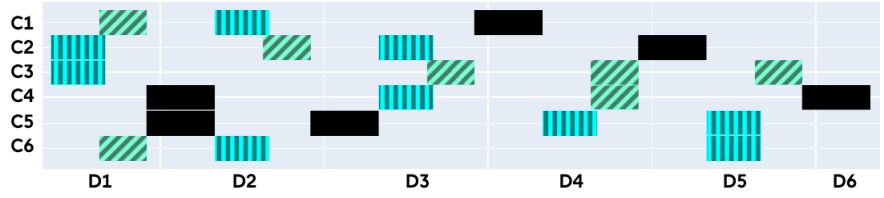

Fig. 3 A construction method is applied using the timetable depicted in Figure 2.

The side length parameter ϕ needs to be tuned to obtain a good performance of the adaptive mechanism for the Smart Destruction method and for the BIA construction method.

Finally, $ChooseConstructionMethod()$ makes use of a parameter ψ for choosing either the BIA (with a probability ψ) or the version of the ILPH (with a probability $1 - \psi$). The parameter ψ needs to be tuned to obtain a good trade-off between the use of the BIA or the use of the construction method based on the ILPH.

Acceptance strategy

The acceptance strategy we implemented in $AcceptDegradation(S_{cur}, S_{old})$ is based on the record-to-record approach introduced in Dueck (1993). Provided that $AcceptIter$ iterations has been performed without improving the quality of S_{old}, this solution may be replaced by S_{cur} a solution of poorer quality. We make use of an acceptance rate τ to be tuned that plays the role of a deviation parameter. The solution S_{cur} takes place of S_{old} if $(S_{cur} - S_{old})/S_{old} \leq \tau$.

7 Computational experiments

In our experiments, our objectives were: (i) to provide a summary on the tuning the parameters of the ALNS metaheuristic that makes it possible to obtain the best results; (ii) to show the efficiency of the adaptive mechanisms we implemented for the destruction and the construction methods; (iii) to evaluate the contribution of the destruction methods; (iv) to compare performances between the ILP model, the ILPH matheuristic and two versions of the ALNS approach; (v) to assess the quality of the solutions computed by the ILP, the ILPH and the ALNS on the instances of the datasets compared to previous approaches.

Tests were done using C++ compiled with gcc version 7.5.0, using STL, using a CPLEX 12.10 IBM (2020) solver with a single thread and the ILPEmphasis parameter set to feasibility, on a machine with an Intel(R) Xeon(R) X7542 CPU @ 2.6 GHz and 64 GB of RAM.

Datasets overview and weights of the terms of the objective function

We tested the ILP, the ILPH and the ALNS versions on a benchmark composed of four datasets of seven instances that we generated using real data of the INFOCA firefighter institution [1]. The instances in datasets are ranged and labelled according to the number of crews n_c and to the total daily number of working shift demands (i.e. $\sum r_s$). So, instances are denoted as $cXXrYY(a/b)$, the (a/b) notation is used whether n_c and $\sum r_s$ equals for two distinct instances which are different in minimum demands distributions.

According to the requirements of the INFOCA institution and to express the relative importance of the different constraints, we set w_{oc} to 2, w_{sg} to 1,

[1] https://datasets.hds.utc.fr/project/10

w_{sst} to 1, w_{ca} to 1 and w_p to 2. These weights are used to compute $Obj(S)$, the objective function value (see Equations (1a)-(1e)).

Parameters tuning

The ALNS that we propose makes use of parameters that need to be tuned to obtain a good efficiency of destruction, construction, adaptive mechanisms and acceptance procedures. We overview these parameters in Table 2.

Table 2 Parameters ϕ, ε, λ, τ, ψ and $Size_w$ to be tuned for the ALNS for FFTP.

ϕ	side length, adaptive mechanism for parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$
ε	$MaxIter = \varepsilon.n_c$, maximum number of iterations without any improvement
λ	adaptive mechanism for managing destruction methods
τ	acceptance rate
ψ	probability used for choosing from among construction methods
$Slide_w$	by how many days the window slides (ILPH)

We carried out preliminary experiments in order to calibrate the parameters. Two instances are selected from each dataset at random for tuning. Since ALNS is a randomized search method, the experiments were repeated ten times with a different random seed.

We evaluate the tuning of the parameters using the **R**elative **P**ercentage **E**rror metric computed as $RPE = 100 \cdot \frac{Z_{min} - Z_{best}}{Z_{best}}$, where Z_{best} denotes the best result we obtained over all the performed runs for an instance, and Z_{min} denotes the best result we obtained among the ten performed runs. The **RPE** values are averaged and reported as a percentage in the figures.

We started by tuning the first ϕ, and we then tune each parameter one after the other considering the order of Table 2. The initial values we use to first tune ϕ are $\varepsilon = 5$, $\lambda = 1$, $\tau = 0.01$, $\psi = 0.5$ and $Slide_w = 3$. Then, to tune a parameter we retain the best setting of the other parameters found before proceeding to tune it.

The value of ϕ, the side length for the adaptive construction mechanism, may affect significantly the performances of the ALNS. This mechanism makes it possible to adapt the values of $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ parameters so as to the relative importance of the terms of BIDC may change during the course of the algorithm.

In Figure 4, we show the impact on the average RPE by varying ϕ from 0.0 to 1.0 with a step of 0.025. When ϕ is lower than 0.1, the four parallel independent searches use values of $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ too closed to the initial one. When ϕ is larger than 0.1, the average RPE is worsened, as can be seen by the

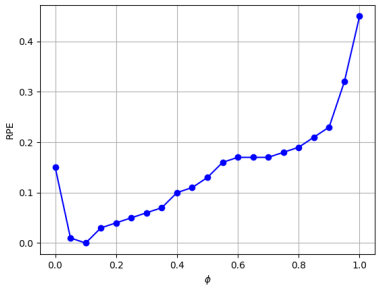

Fig. 4 Tuning of ϕ, effect on average RPE.

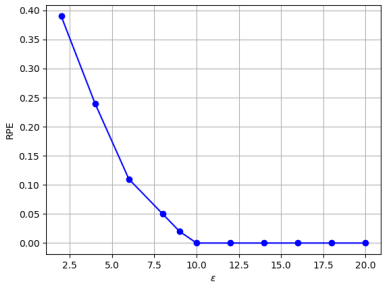

Fig. 5 Tuning of ε, effect on average RPE.

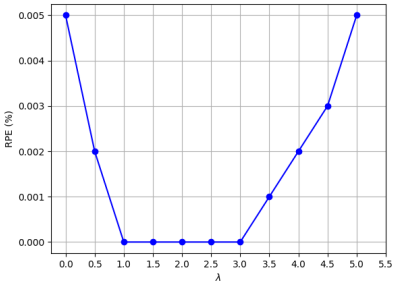

Fig. 6 Tuning of λ, effect on average RPE.

average RPE values. We chose to set ϕ at 0.1 for which the minimum average RPE is obtained.

The maximum number of iterations without any improvement is $MaxIter = \varepsilon.n_c$ where n_c is the number of crews. As could be expected, the processing time increases with the ε value. As would be expected with this tuning, the average RPE decreases as ε increases. To tune ε, we vary it in $\{2, \cdots, 20\}$. As can be seen in Figure 5, the minimum is attained when ε is equal to ten, then it becomes constant. We chose to set ε to ten so as not to waste processing time.

The **De**struction **Sc**ore of method is managed by the parameter λ. In Figure 6, we show the evolution of the average RPE by varying λ from 0 to 5 with a step of 0.5. When the value $\lambda = 0$ the score update mechanism is disabled, and we do not obtain the minimum average RPE but the value is closed to zero. The RPE values decrease until $\lambda = 1.0$, then they are constant for λ values in $[1, 3]$. Next, RPE values increase as the λ value increases. The score update mechanism makes it possible to obtain better results by managing the selection of the destruction methods. We chose to set λ to =1.0, this suffices to make the mechanism effective.

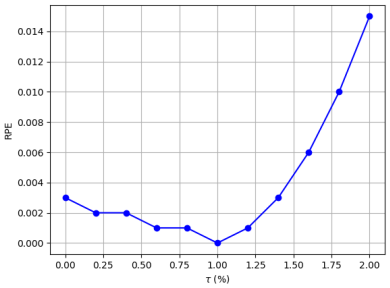

Fig. 7 Tuning of τ, effect on average RPE.

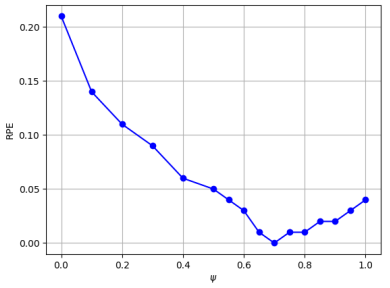

Fig. 8 Tuning of ψ, effect on average RPE.

A solution can be replaced by a solution of poorer quality when $(S_{cur} - S_{old})/S_{old} \leq \tau$. In Figure 7, we show the evolution of the average RPE by varying τ from 0 % to 2 % with a step of 0.2 %. When $\tau = 0\,\%$, the record-to-record acceptance mechanism is disabled. As it can be seen, we obtain a good average RPE value with $\tau = 0\%$. The RPE values decrease until the value $\tau = 1\%$, next they increase as τ increases. As it could be expected, the acceptance mechanism based on the record-to-record algorithm plays its role. We chose to set $\tau = 1\%$.

The BIA is chosen with a probability ψ and the ILPH is chosen with a probability $1 - \psi$. In Figure 8, we show the evolution of the average RPE by varying ψ from 0.0 to 1.0 with a step of 0.1. When $\psi = 0.0$ the ALNS only uses the ILPH as a construction method, this is not efficient as can be seen by the average RPE value. The RPE values decreases up to $\psi = 0.7$, next they increase as ψ increases The ILPH contributes to obtain better results that can be observed in Figure 8. We chose $\psi = 0.7$, which is a good trade-off between the construction methods.

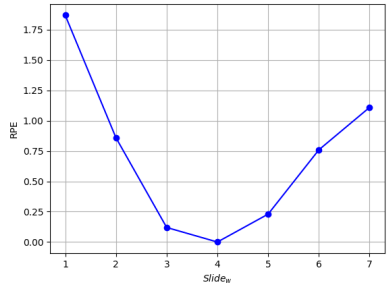

Fig. 9 Tuning of $Slide_w$, effect on average RPE.

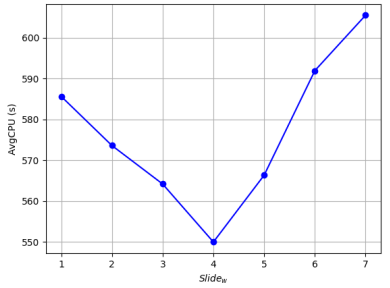

Fig. 10 Tuning of $Slide_w$, effect on processing time.

We use a version of the ILPH as a construction method with only the Fix Day neighborhood. We performed experiments by varying $Slide_w$ in $\{1, 7\}$,

and with one minute for the time limit T_{ILPH}. In Figure 9, we show that when $Slide_w$ varies between one and four the RPE value decreases, and in Figure 10 we show that the processing time decreases. Next, the average RPE and the processing time values increase as $Slide_w$ increases. When $Slide_w < 4$ the windows are too small that does not allow the Fix Day neighborhood to find better solutions than the ones found by BIA. We waste processing time running the construction method based on ILPH without any benefit. When $Slide_w > 4$, the construction method based on ILPH requires more processing time to obtain solutions. This also negatively impacts the convergence of the algorithm towards good solutions since we set a time limit in order to avoid stucking in a hard ILP resolution. We chose to set $Slide_w$ to four.

The final calibration values are $\phi = 0.1$, $\varepsilon = 10$, $\lambda = 1.0$, $\tau = 0.01$, $\psi = 0.7$ and $Slide_w = 4$. They were used in the sequel for our experiments on all the benchmark instances, and they were chosen to obtain a good trade-off between solution quality and processing time.

Evaluation of the ILPH neighborhoods

We evaluate here the effectiveness of the proposed neighborhoods for the ILPH method. In Table 3 under the "ILPH-Sequ" heading, we show the results with the three neighborhoods used one after the other. Next, under the "No-FD", "No-FS" and "No-FC" headings we show the results by disabling each of them one at a time. Under the "ILPH" heading, we show the results of the proposed ILPH we detailed in Algorithm 1. For the sake of compactness, we computed the average RPE by datasets. For each of the tested versions of ILPH and for each instance, the computing time limit is set to 3600 seconds. The best results are shown in bold print.

Table 3 Evaluation of the ILPH neighborhoods, average RPE values by dataset.

	ILPH-Sequ	No-FD	No-FS	No-FC	ILPH
c18	0.13	0.23	0.16	0.15	**0.06**
c30	0.15	0.26	0.19	0.19	**0.07**
c50	0.21	0.32	0.25	0.26	**0.11**
c70	0.29	0.47	0.36	0.38	**0.17**

We can see in Table 3 that the ILPH version without Fix Day neighborhood obtains the worst average RPE values compare to the versions without Fix Shift or Fix Crews neighborhoods. This means that the Fix Day neighborhood is more efficient that the two others.

We can also see that ILPH-Sequ version, which runs the three neighborhoods at each iteration obtains poorer results than the ILPH version of Algorithm 1. We experimentally observed that running all neighborhoods in turn at each iteration is time consuming and slow the convergence towards good solutions.

Hence, we decided to use a hierarchical structure for the ILPH method, where the most efficient neighborhood (Fix Day) is used at each iteration, and when a better solution is found, we run the other neighborhoods to try to improve this solution.

Evaluation of the ALNS adaptive mechanisms, evaluation of the destruction methods

We evaluate here the effectiveness of the BIA adaptive mechanism and the effectiveness of the diversification mechanism for destruction. We also evaluate the effectiveness of the destructions methods by disabling each of them one at a time. For these experiments we use all the instances of the four datasets and we record the best solution found over 15,000 iterations. We performed ten runs on each benchmark instances.

The parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ is used to compute the BIDC. The adaptive construction mechanism aims to guide the search by computing the best trade-off between the different value of these parameters over consecutive iterations. We conducted experiments with the adaptive construction mechanism and without the adaptive construction mechanism. In that latter case, the parameters $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ are chosen randomly in $[0, 1]$. In Figure 11 we show the average RPE vs. iterations for these two versions.

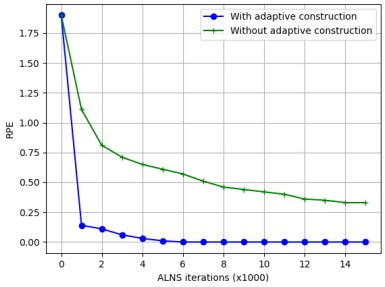

Fig. 11 Adaptive BIA construction impact.

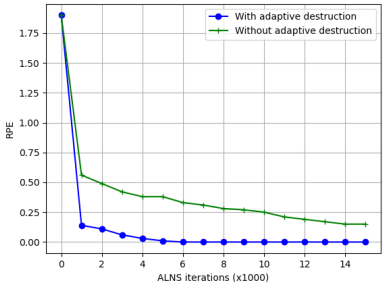

Fig. 12 Adaptive destruction impact.

As can be seen in Figure 11, the RPE values are worsened when parameters $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ are chosen at random. The adaptive construction mechanism with ϕ set at 0.1 significantly speed-up the convergence towards good solutions.

The adaptive diversification mechanism manages the value of $D_{max} \in \{3, \cdots, D_{limit}\}$ (see Algorithm 2). This mechanism makes it possible to explore in the vicinity of the new solution as soon as an improvement is found and also makes it possible to explore more distant neighborhood whenever the search is trapped in a local optimum. We disabled the adaptive diversification mechanism by fixing the D_{max} diversification degree to 3. In Figure 12, we show the average RPE vs. iterations for these two versions. The adaptive diversification

mechanism always obtains better results as can be seen by the average RPE values. The adaptive diversification mechanism makes it possible to converge faster towards good solutions.

In Table 4, we show the results for the destruction methods. For the sake of comparison, we use the Average Relative Percentage Error computed as $ARPE = 100 \cdot \frac{Z_{avg} - Z_{best}}{Z_{best}}$ where Z_{best} denotes the best result we obtained over all the runs we performed for an instance, and Z_{avg} denotes the average of the results. This allows us to discuss the results to show whether the ALNS is stable over the runs. We also use the Relative Percentage Gap criterion computed as $RPG = 100 \cdot \frac{Z_{min} - Z_{ILP}}{Z_{ILP}}$ where Z_{ILP} denotes the value attained by the ILP for an instance (when possible), and Z_{min} denotes the best result obtained among the performed runs. This allows us to show how far from optimal values the best solutions computed by ALNS are.

In Table 4 under the "ALNS" heading, we show the results with RD and SD activated. Under the "No RD" and "No SD" headings we show the results by disabling each of them one at a time. For the sake of compactness, we computed the average ARPE and RPG values by datasets. The average processing time is in seconds. The best results are shown in bold print, and **nc** stands for **n**ot **c**alculable.

Table 4 Evaluation of the destruction methods.

	ALNS			No RD			No SD		
	ARPE	RPG	t(s)	ARPE	RPG	t(s)	ARPE	RPG	t(s)
c18	**0**	**0**	**323.42**	**0**	**0**	331.45	0.56	1.71	542.31
c30	**0.06**	**0**	**494.14**	0.09	**0**	511.26	0.89	3.93	788.23
c50	**0.20**	nc	**597.28**	0.25	nc	639.43	1.14	nc	923.67
c70	**0.43**	nc	**842.57**	0.51	nc	877.54	1.42	nc	1,269.03

The ALNS without Random Destruction ("No RD" heading) obtains results for ARPE that are below those of ALNS that uses RD and SD, and the processing times are greater. The Random Destruction makes it possible to avoid being stuck in a local optimum by performing perturbations.

The ALNS without Smart Destruction ("No SD" heading) obtains results for ARPE and RPG criteria that are below those of ALNS that uses RD and SD, and the processing times are greater. As can be seen in Table 4 by comparing the "t(s)" columns, the Smart Destruction speeds up the convergence of the ALNS algorithm and allows us to obtain better solutions.

The RPG values can be used for comparison for the c18 and c30 datasets for which optimal solutions can be obtained.

The SD method is efficient, we obtain optimal solutions for these datasets. As it can be observed for c50 and c70 datasets the RD method is needed since better results are attained in shorter processing times. The Random Destruction and the Smart Destruction methods are beneficial for exploring the neighborhood of a solution either by obtaining better results or by shortening the processing time.

Comparison of ILP, ILPH and ALNS

All of the proposed solution approaches are run within a time limit set at 3,600 s. For the experiments conducted with ILPH, the initial solution is obtained using $BuildFeasibleSchedule()$, T_{ILPH} is set at 3,600 s and T_l is set at 30 s.

For comparison purposes, we decided to conduct experiments with two versions of the ALNS, without and with the constructive method based on ILPH. To shorten the processing time, we only use the Fix Day neighborhood as a constructive method with $Size_w = 8$, $Slide_w = 4$ and $T_l = 60s$. When we performed our preliminary experiments, we found that these values are a good trade-off between expected gain and processing time. We run the two ALNS versions ten times on each instance.

We use the Relative Percentage Gap (RPG) and Average Relative Percentage Error (ARPE) as assessment criteria.

In Table 5, we show the results for the c18 and the c30 datasets for which the ILP succeeded in obtaining a solution. In Table 6, we show the results for the c50 and the c70 datasets for which the ILPH succeeded in obtaining a solution when the ILP failed in attaining a solution within the time limit. In these tables, **ns** stands for **n**o **s**olution, **nc** stands for **n**ot **c**alculable, and **-** shows that the time limit is attained. The last row of the two tables reports some average values. The best results are shown in bold print.

Table 5 For c18 and c30 datasets, results of ILP, ILPH matheuristic, ALNS without the ILPH and ALNS.

	ILP			ILPH			ALNS (no ILPH)				ALNS			
	Obj	gap	t(s)	Obj	gap	t(s)	Z_{min}	RPG	ARPE	t(s)	Z_{min}	RPG	ARPE	t(s)
c18r09a	**1,325**	0	1,023	1,432	8.08	-	**1,325**	0	0	271	**1,325**	0	0	281
c18r10a	**1,359**	0	1,001	1,456	7.14	-	**1,359**	0	0	287	**1,359**	0	0	292
c18r10b	**1,344**	0	1,112	1,421	5.73	-	**1,344**	0	0	301	**1,344**	0	0	311
c18r11a	**1,378**	0	1,234	1,456	5.66	-	**1,378**	0	0	309	**1,378**	0	0	323
c18r11b	**1,420**	0	1,787	1,532	7.89	-	**1,420**	0	0	321	**1,420**	0	0	341
c18r12a	**1,422**	0	1,455	1,498	5.34	-	**1,422**	0	0	341	**1,422**	0	0	355
c18r12b	**1,440**	0	1,564	1,562	8.47	-	**1,440**	0	0	344	**1,440**	0	0	361
c30r15a	**1,754**	0	2,892	1,931	10.09	-	**1,754**	0	0	451	**1,754**	0	0	455
c30r16a	**1,780**	0	2,911	1,945	9.27	-	**1,780**	0	0.04	463	**1,780**	0	0.03	469
c30r17a	**1,810**	0	3,002	1,911	5.58	-	**1,810**	0	0.10	472	**1,810**	0	0.11	483
c30r18a	**1,832**	0	3,014	1,934	5.57	-	**1,832**	0	0.03	481	**1,832**	0	0.04	495
c30r19a	**1,880**	0	3,111	1,990	5.85	-	**1,880**	0	0.05	489	**1,880**	0	0.12	501
c30r20a	1,932	1.79	-	1,978	4.21	-	**1,898**	-1.75	0.13	513	**1,898**	-1.75	0.09	522
c30r21a	**1,913**	0	3,213	2,005	4.81	-	**1,913**	0	0.07	502	**1,913**	0	0.05	534
Avg		0.13	2,208		6.69	3,600		-0.13	0.03	396		-0.13	0.03	408

For each instance, under the "ILP" and the "ILPH" headings, we show the objective function value, the gap and the processing time (in seconds) that we obtained for the ILP and the ILPH matheuristic (under "Obj", "gap" and "t(s)" columns, respectively).

For each instance, under the "ALNS (no ILPH)" and the "ALNS" headings, we show the best result obtained among the performed runs (the best Obj value), the Relative Percentage Gap, the Average Relative Percentage Error

and the computing time in seconds (under "Z_{min}", "RPG", "ARPE" and "t(s)" columns, respectively).

In Table 5, we show that the ILP attains optimal solutions for all the instances except for the c30r20a instance for which the gap is 1.79%. None of the optimal solutions are obtained by the ILPH matheuristic, and the average gap is 6.69%. The two versions of the ALNS succeeded in obtaining all the optimal solutions. A better solution is found for the c30r20a instance when the ILP cannot attain an optimal solution within the one hour time limit.

The RPG and ARPE values are either equal to zero or to almost zero except for the c30r20a instance. For the c30r20a instance, the two ALNS versions succeeded in attaining a better solution, we therefore obtain the negative value −1.75% for RPG. When we used the solutions found by the two versions of the ALNS for this instance as initial solutions for the ILP model, we found that they are optimal ones. For the c30 dataset, the two ALNS versions attain all of the optimal solutions.

For the c18 and c30 datasets, the ALNS versions are stable when computing solutions, as can be seen by the ARPE values. The processing times of the ALNS versions are about five time shorter than those of the ILP in average, and for all the instances we obtained optimal solutions.

Table 6 For c50 and c70 datasets, results of ILP, ILPH matheuristic, ALNS without the ILPH and ALNS.

	ILP			ILPH			ALNS (no ILPH)				ALNS			
	Obj	gap	t(s)	Obj	gap	t(s)	Z_{min}	RPG	ARPE	t(s)	Z_{min}	RPG	ARPE	t(s)
c50r22a	ns	nc	-	3,876	nc	-	**3,623**	nc	0.22	534	**3,623**	nc	0.17	566
c50r23a	ns	nc	-	3,845	nc	-	**3,680**	nc	0.14	545	**3,680**	nc	0.15	579
c50r26a	ns	nc	-	3,854	nc	-	**3,710**	nc	0.15	567	**3,710**	nc	0.23	543
c50r28a	ns	nc	-	3,911	nc	-	**3,740**	nc	0.33	573	**3,740**	nc	0.12	567
c50r31a	ns	nc	-	3,967	nc	-	3,834	nc	0.27	588	**3,830**	nc	0.23	591
c50r33a	ns	nc	-	4,034	nc	-	**3,878**	nc	0.31	579	**3,878**	nc	0.29	563
c50r35a	ns	nc	-	4,211	nc	-	**3,987**	nc	0.21	781	**3,987**	nc	0.22	772
c70r31a	ns	nc	-	4,931	nc	-	**4,609**	nc	0.39	793	**4,609**	nc	0.39	832
c70r33a	ns	nc	-	4,976	nc	-	**4,680**	nc	0.43	814	**4,680**	nc	0.41	801
c70r37a	ns	nc	-	5,011	nc	-	**4,713**	nc	0.47	803	**4,713**	nc	0.43	844
c70r40a	ns	nc	-	4,988	nc	-	**4,770**	nc	0.52	841	**4,770**	nc	0.50	866
c70r44a	ns	nc	-	5,113	nc	-	**4,834**	nc	0.37	857	**4,834**	nc	0.41	811
c70r47a	ns	nc	-	5,134	nc	-	**4,936**	nc	0.58	863	**4,936**	nc	0.51	867
c70r50a	ns	nc	-	5,178	nc	-	**5,002**	nc	0.34	879	**5,002**	nc	0.35	877
Avg									0.34	715			0.32	720

In Table 6, we show that the ILP failed in attaining a solution within the time limit, and the ILPH matheuristic succeeded in obtaining feasible solutions. The solution quality of the ALNS versions cannot be assessed using the RPG criterion since the ILP cannot obtain optimal solutions within the time limit.

Except for the c50r31a instance, the two ALNS versions obtained the same Z_{min} values. The computing times are closely the same (about 715 s and 720 s in average), and are at a reasonable extra expense compared to the c18 and

c30 datasets with smaller instances (about 400 s in average). The two ALNS versions remain stable as can be seen by the ARPE values that are closely the same (0.34 and 0.32 in average).

For the c50r31a instance, the ALNS that uses the constructive method based on ILPH succeeded in attaining a better solution. The constructive method based on ILPH contributes to obtain better results as shown by Figure 8 that we presented when tuning of ψ.

The results that we obtained show that the ALNS obtained optimal results on the smaller instances. For the larger instances, the ALNS obtained results of better quality within shorter processing time compared with the ones obtained by the ILPH matheuristic used on its own.

Comparison with the literature

For the sake of compactness the results obtained in Ouberkouk et al. (2021) have not been presented for each instance in Table 5 and Table 6. Instead, in Table 7 and Table 8, we compare the solution approaches by dataset.

The ILP we presented in Section 4 improves the formulations of the constraints that relate to the variables α, β and γ so as to reduce the number of these variables. We now have for each of them of the order of $n_c.n_d$ variables whereas the previous formulations make use of the order of $n_c.n_s.n_s.n_d$ variables. In Table 7, for each dataset, the "#Variables" column shows the sum of the number of variables α, β and γ, the "Avg Gap" column shows the average gap, the "Avg t(s)" column shows the average processing time in seconds, and the "#OS" column shows the number of optimal solutions found.

As can be seen in the '#Variables" columns, the number of variables is reduced. Thus, we obtained six new optimal solutions for dataset c30 whereas no optimal solution was found by the earlier ILP. Moreover, the processing times are reduced.

Table 7 Earlier ILP versus new ILP.

	Earlier ILP				New ILP			
Dataset	#Variables	Avg Gap	Avg t(s)	#OS	#Variables	Avg Gap	Avg t(s)	#OS
c18	129,360	0	1,488	7/7	2,640	0	1,311	7/7
c30	216,090	2.67	3,600	0/7	4,410	0.25	3,106	6/7
c50	360,150	nc	3,600	nc	7,350	nc	3,600	nc
c70	504,210	nc	3,600	nc	10,290	nc	3,600	nc

Table 8 AIDCH versus ALNS.

	AIDCH				ALNS			
Dataset	Avg Z_{min}	Avg RPG	Avg ARPE	Avg t(s)	Avg Z_{min}	Avg RPG	Avg ARPE	Avg t(s)
c18	1,384	0	0	374	1,384	0	0	323
c30	1,873	5.97	0.16	602	1,838	0	0.06	494
c50	3,878	nc	0.43	791	3,778	nc	0.20	597
c70	5,185	nc	0.70	1,038	4,792	nc	0.43	843

In Table 8, we show a comparison between the Adaptive Iterative Destruction Construction Heuristic (AIDCH) presented in Ouberkouk et al. (2021) and the ALNS that we proposed. The "Avg Z_{min}" column shows the average of the best results obtained among the performed runs, the "Avg RPG" and the "Avg ARPE" columns show the average RPG and average ARPE when they can be computed, and the "Avg t(s)" column shows the average processing time in seconds.

Equivalent (c18) or better results (c30, c50 and c70) are obtained by the ALNS, as an be seen by the Avg Z_{min} values. The ALNS is more stable as can be seen by the average ARPE values. Moreover, the processing times are reduced. The ALNS performs better than the AIDCH solution approach for all of the metrics we used.

The AIDCH does not make use of multiple destruction/construction methods together with the associated adaptive mechanisms. The AIDCH does not implement an acceptance mechanism of low quality solution. Moreover, the improvements of the ILP makes it possible to propose a version of the ILPH that we used as a construction method for the ALNS.

8 Conclusion and future work

In this paper, we addressed the real-world firefighters timetabling problem (FFTP) of the INFOCA institution for which firefighter crews have to be scheduled within the yearly forest fire period. The issue faced is to obtain a maximal operational capacity while considering the constraints that make it possible to achieve this operational capacity, in addition to the constraints related to work regulation. We presented an ILP model, an ILPH matheuristic and an ALNS metaheuristic to address the FFTP of INFOCA. The proposed approaches were tested using four datasets of increasing difficulty in size that we generated using real data.

The ILP approach obtained optimal results for the two datasets with small instances but faced difficulty in obtaining feasible solutions for the larger instances within reasonable processing time. The ILP model allowed us to design the ILPH matheuristic that makes it possible to obtain solutions for all of the instances but of lower quality compared to the optimal solutions that we obtained, and at the expense of large processing time. To achieve better results within shorter processing times we then proposed an ALNS metaheuristic solution approach that makes use of both the Best Insertion Algorithm and of a version of the ILPH matheuristic as constructive methods. Computational experiments were conducted to tune the parameters to obtain a good trade-off between solution quality and processing time. Additional computational experiments were conducted to evaluate the effectiveness of the main components of the ALNS approach, and all these components are necessary to obtain good solutions within good processing time. The ALNS obtained all of the optimal results that were obtained using the ILP on the smaller instances. For the larger instances, the results are of better quality compared with those obtained

by the ILPH matheuristic, in addition the processing times are significantly reduced.

A future research direction would be to consider a broarder problem by considering the schedule of preventive maintenance operations of material resources, these may impact the availability of some resources thus the number of some types of shift per day.

Acknowledgments

This work was initiated as part of the GEO-SAFE project. The GEO-SAFE project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie RISE grant agreement No 691161. This work is carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Afshar-Nadjafi, B. (2021). Multi-skilling in scheduling problems: A review on models, methods and applications. *Computers & Industrial Engineering*, 151:107004.

Brucker, P., Qu, R., and Burke, E. (2011). Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3):467–473.

Burke, E. K., De Causmaecker, P., Berghe, G. V., and Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of scheduling*, 7(6):441–499.

Chandrasekharan, R. C., Smet, P., and Wauters, T. (2021). An automatic constructive matheuristic for the shift minimization personnel task scheduling problem. *Journal of Heuristics*, 27(1-2):205–227.

Curtois, T. (2014). Employee shift scheduling benchmark data sets.

De Bruecker, P., Beliën, J., Van den Bergh, J., and Demeulemeester, E. (2018). A three-stage mixed integer programming approach for optimizing the skill mix and training schedules for aircraft maintenance. *European Journal of Operational Research*, 267(2):439–452.

De Bruecker, P., Van den Bergh, J., Beliën, J., and Demeulemeester, E. (2015). Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16.

Demirović, E., Musliu, N., and Winter, F. (2019). Modeling and solving staff scheduling with partial weighted maxsat. *Annals of Operations Research*, 275(1):79–99.

Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, 104(1):86–92.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. (2004a). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1):21–144.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004b). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27.

Fages, J.-G. and Lapègue, T. (2014). Filtering atmostnvalue with difference constraints: Application to the shift minimisation personnel task scheduling problem. *Artificial Intelligence*, 212:116–133.

Guerriero, F. and Guido, R. (2022). Modeling a flexible staff scheduling problem in the Era of Covid-19. *Optimization Letters*, 16(4):1259–1279.

Heil, J., Hoffmann, K., and Buscher, U. (2020). Railway crew scheduling: Models, methods and applications. *European journal of operational research*, 283(2):405–425.

Hoffmann, K. and Buscher, U. (2019). Valid inequalities for the arc flow formulation of the railway crew scheduling problem with attendance rates. *Computers & Industrial Engineering*, 127:1143–1152.

Hojati, M. (2018). A greedy heuristic for shift minimization personnel task scheduling problem. *Computers & Operations Research*, 100:66–76.

Hussain, K., Mohd Salleh, M. N., Cheng, S., and Shi, Y. (2019). Metaheuristic research: a comprehensive survey. *Artificial intelligence review*, 52(4):2191–2233.

IBM (2020). Cplex User's Manual.

Kletzander, L. and Musliu, N. (2020). Solving the general employee scheduling problem. *Computers & Operations Research*, 113:104794.

Krishnamoorthy, M., Ernst, A. T., and Baatar, D. (2012). Algorithms for large scale shift minimisation personnel task scheduling problems. *European Journal of Operational Research*, 219(1):34–48.

Lapègue, T., Bellenguez-Morineau, O., and Prot, D. (2013). A constraint-based approach for the shift design personnel task scheduling problem with equity. *Computers & Operations Research*, 40(10):2450–2465.

Mara, S. T. W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., and Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, page 105903.

Ouberkouk, M.-A., Boufflet, J.-P., and Mourkim, A. (2021). Adaptive iterative destruction construction heuristic for the firefighters timetabling problem. In *8th International Conference on Metaheuristics and Nature Inspired Computing*.

Porto, A. F., Henao, C. A., López-Ospina, H. A., and González, E. R. (2019). Hybrid flexibility strategy on personnel scheduling: Retail case study. *Comput. Ind. Eng.*, 133:220–230.

Pour, S. M., Drake, J. H., Ejlertsen, L. S., Rasmussen, K. M., and Burke, E. K. (2018). A hybrid constraint programming/mixed integer programming framework for the preventive signaling maintenance crew scheduling

problem. *European Journal of Operational Research*, 269(1):341–352.

Qin, R., Nembhard, D. A., and Barnes II, W. L. (2015). Workforce flexibility in operations management. *Surveys in Operations Research and Management Science*, 20(1):19–33.

Santos, H. G., Toffolo, T. A., Gomes, R. A., and Ribas, S. (2016). Integer programming techniques for the nurse rostering problem. *Annals of Operations Research*, 239(1):225–251.

Smet, P., Wauters, T., Mihaylov, M., and Berghe, G. V. (2014). The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega*, 46:64–73.

Tadumadze, G., Boysen, N., Emde, S., and Weidinger, F. (2019). Integrated truck and workforce scheduling to accelerate the unloading of trucks. *European Journal of Operational Research*, 278(1):343–362.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European journal of operational research*, 226(3):367–385.

Zucchi, G., Iori, M., and Subramanian, A. (2021). Personnel scheduling during Covid-19 pandemic. *Optimization Letters*, 15(4):1385–1396.