Adaptive iterative destruction construction heuristic for the firefighters timetabling problem

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

Adaptive iterative destruction construction heuristic for the firefighters timetabling problem

Mohamed-Amine Ouberkouk, Jean-Paul Boufflet and Aziz Mourkim

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex
{mohamed-amine.ouberkouk,jean-paul.boufflet,aziz.moukrim}@hds.utc.fr

Abstract. Every year, wildfires accentuated by global warming, cause economic and ecological losses, and often, human casualties. Increasing operating capacity of firefighter crews is of importance to better face the forest fire period that yearly occurs. In this study, we investigate the real-world firefighters timetabling problem (FFTP) of the INFOCA institution in Andalusia (Spain) with the aim of increasing operating capacity while taking into account work regulation constraints. We propose an Integer Linear Programming model and an Adaptive Iterative Destruction Construction Heuristic solution approache to address the problem. 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[1].

Keywords. Timetabling, Firefighters, ILP, Adaptive destruction/contruction heuristic

1 Introduction

Timetabling problems [1, 7, 9] involve allocating resources within time slots considering a predefined planning horizon while respecting precedence, duration, capacity, disjunctive and distribution (spacing, grouping) constraints. Staff planning aims at building timetables so that an organization can meet demands for goods or services. For each staff member, working and rest days are scheduled in a timetable while taking into account work regulation constraints and local regulation constraints, if any.

The first works on personnel scheduling can be traced back to Edie's work on traffic delays at toll booths [5]. Since then, scheduling algorithms have been applied in a lot of areas like transportation systems (airlines, railways), healthcare systems, emergency services (police, ambulances), call centers and other services (hotels, restaurants, commercial stores).

Comprehensive literature reviews covering a wide area of problems with many references on personnel scheduling can be found in [6, 10]. The works are classified by type of problem, application area and solution method. As an example, the nurse rostering [4] is a scheduling issue in health systems. The objective is to build a daily schedule for nurses with the aim of obtaining a full timetable over few weeks for the institution. The rosters should provide suitably qualified nurses to cover the demand of working shifts arising from the numbers of patients in the wards. The resulting schedule should comply with regulatory constraints and should ensure that night and weekend shifts are fairly distributed while accommodating nurse preferences.

Staff scheduling is known as crew scheduling in transportation systems areas such as market/airlines, railways, mass transit and buses [2]. For these problems, there are two common features. The first is that both temporal and spatial constraints are involved. Each task is characterized by its starting time and location, and, its ending time and location. The second is that all tasks to be performed by employees are determined from a given timetable. The tasks are determined following a decomposition of the different duties that the company must ensure within a planning period. A task may be assuring a flight leg in airlines or ensuring a trip between two segments in a train.

The firefighters problem that we address consists in providing the INFOCA's daily schedule within a fixed planning horizon for a number of firefighter crews. Each firefighter is assigned to a crew for a year. These firefighters crews can be assigned to several types of shifts such as helicopter

[1] https://geosafe.lessonsonfire.eu/

work, night work, work on demand (24 hour on call). The planning period is the high-risk period from 1st June to 15th October where wildfires yearly occur (forest fire period).

The objective is to build a schedule for every crew of firefighters, hence a full timetable that covers all the forest fire period. The aim is to maximize the overall operating capacity while respecting the minimum demands for each shift, the regulatory constraints imposed by the institution as well as other soft constraints of good practice in order to make the schedules adequate to the preferences of the institution. The constraints of good practice relate to the grouping of assignments of same shifts within consecutive days, the allocation of compensations after rest days while maximizing of the number of operational crews a day.

The application of various metaheuristics to employee scheduling problems is presented in the reviews mentioned above. In this study, we choose to investigate an algorithm mainly based on an Adaptive Iterative Destruction/Construction Heuristic (AIDCH) [3]. An initial feasible solution that only complies with the minimum demands is build first by applying a constructive heuristic. Then, the AIDCH approach that we propose aims at increasing the overall operating capacity by first partly destroying a solution, next it is completed by inserting as many crews as possible, that can be easily done through a Destruction/Construction Heuristic approach. While completing the solution to increase the overall operational capacity, we make work together adaptive diversification mechanisms and parallel independent searches to avoid to be trapped in a local optimum.

In this paper we propose an Integer Linear Programming (ILP) formulation together with an Adaptative Iterative Destruction Construction Heuristic (AIDCH) to address the firefighters timetabling problem (FFTP) of the INFOCA institution. The ILP is designed for modeling purposes and with the aim of giving lower bounds useful for the tuning analysis of the AIDCH solution approach. The Adaptive Iterative Destruction/Construction Heuristic is composed of an adaptive diversification mechanism at the destruction phase followed by an adaptive construction phase, based on a Best Insertion Algorithm, which performs parallel independent searches. The initial parameter values are adjusted by the algorithm according to the solution progress throughout the resolution process. The AIDCH is appropriate to generate solutions of good quality for the larger instances. The remainder of the paper is organized as follows. Section 2 provides a description of the FFTP, then the ILP formulation is presented in Section 3. The proposed AIDCH solution approach is described in Section 3. Computational experiments performed on a benchmark that we generated using real data of the INFOCA firefighter institution are reported in Section 4. Conclusion and future works are given in Section 5.

2 Problem description

In this section we present a global overview of the real-world firefighter planning problem that we address. We gives 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 notations used for the types of shifts and their brief descriptions are the following:

(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.

For the considered firefighters timetabling problem, the hard constraints relating to work regulation and to local regulation of the INFOCA institution are the following:

(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 have a maximum number of consecutive working days.

Some consecutive shift assignment are forbidden for a crew (H3), for instance a night shift ends at 8 am and cannot be followed by an helicopter shift which begins at 8 am, this forbidden consecutive shift assignment is denoted as (N, H).

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 shifts should be balanced each day. The unbalance on extra assignment to different shifts should be minimized each day.

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 for a day, we had better to assign a crew to three different shifts to balance operating capacity rather than assigning the three crews to a same shift.

3 ILP model for FFTP

In this section we present the ILP model for minimizing the criteria detailed in Section 2. The ILP has a twofold objective, first a modeling purpose for investigating the problem we face, second we aim at obtaining optimal values whether possible for the smaller instances within a reasonable time limit (or lower/upper bounds). This allows to get reference values to make comparisons with the AIDCH solution approach that we propose. We present data and parameters prior to the decision variables, we then give the model.

The data and parameters are the following:

$Days$ set of days of the planning period, a day $d \in [1, \cdots, l_d]$, size n_d;
$Shifts$ set of types of shifts, a shift $s \in \{T12,\ T16,\ H,\ N,\ G7,\ G24,\ A3, R,\ C\}$, size n_s;
$Crews$ set of firefighter crews, size n_c;
l_d last day of the planning period;
F set of couples of forbidden consecutive shift assignment, e.g. $(N, H) \in F$;
r_s daily minimum demand for a working shift $s \in \{Shifts \setminus \{R,\ C\}\}$;
l_s duration of shift s (length in hours);
L maximum workload for any crew over the planning period;
t_s start time of shift s;
w_{oc} **o**perating **c**apacity weight;
w_{sg} shift **g**rouping violation weight (S1);
w_{sst} **s**ame **s**tart **t**ime change violation weight (S2)
w_{ca} compensation **a**ssignments violation weight (S3);
w_p **p**references violation weight (S5);

p_{csd} if crew c does not prefer to work on shift s on day d $p_{csd} = w_p$, zero otherwise (S5);
MAX_d maximum number of consecutive work days for a crew (H6);
WHC number of worked hours giving a compensation day.

The primary boolean variables are X_{csd}, if the crew c works on shift s in day d then $X_{csd} = 1$, zero otherwise. The secondary boolean variables used in the model are the followings:

$\alpha_{css'd} = 1$ if crew c works on shift s in day d and works on a different shift s' in day $d+1$, zero otherwise;
$\beta_{css'd} = 1$ 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;
$\gamma_{css'd} = 1$ 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.

$\alpha_{css'd} = 1$ if a shift change violation occurs (S1, shift grouping), $\beta_{css'd} = 1$ if a working time change violation occurs (S2, same start time) and $\gamma_{css'd} = 1$ if a compensation assignment violation occurs (S3, compensation assignment).

The integer variables used in the model are the followings:

λ_d daily difference between the maximum number of assignable crews (n_c) and those assigned;
δ_c total number of worked shifts for crew c over the planning period;
θ_c total working time of crew c over the planning period;
ρ_{cd} number of worked hours of crew c from the first day to day d;
$\phi_{cc'}$ number of shift assignment difference between the crews c and c' (S4);
$\varphi_{cc'}$ working time difference between the crews c and c' (S4);
$\psi_{ss'}$ unbalance of assignments between the shifts s and s' (S6).

The aim is to maximize operating capacity over the planning period while minimizing the soft constraint violations. We propose the following ILP to address this problem:

Min

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

$$\sum_{c \in Crews} \sum_{s \in Shifts \setminus \{R,\, C\}} \sum_{s' \in Shifts \setminus \{R,\, C\}} \sum_{d \in Days} (w_{sg} \cdot \alpha_{css'd} + w_{sst} \cdot \beta_{css'd} + w_{ca} \cdot \gamma_{css'd}) \tag{1b}$$

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

$$+ \sum_{c \in Crews} \sum_{s \in Shifts \setminus \{R,\, C\}} \sum_{d \in Days} p_{csd} \cdot X_{csd} \tag{1d}$$

$$\sum_{s \in Shifts \setminus \{R,\, C\}} \sum_{s' \in S \setminus \{R,\, C\}} \psi_{ss'} \tag{1e}$$

Subject to:

$$\sum_{s \in Shifts} X_{csd} = 1 \quad \forall c \in Crews,\ \forall d \in Days \tag{2}$$

$$\sum_{c \in Crews} X_{csd} \geq r_s \quad \forall d \in Days,\ \forall s \in \{Shifts \setminus \{R,\ C\}\} \tag{3}$$

$$X_{csd} + X_{cs'(d+1)} \leq 1 \quad \forall (s, s') \in F,\ \forall c \in Crews,\ \forall d \in Days \setminus \{l_d\} \tag{4}$$

$$\sum_{s \in \{Shifts \setminus \{R,\, C\}\}} \sum_{d \in Days} l_s \cdot X_{csd} \leq L \quad \forall c \in Crews \tag{5}$$

$$\sum_{s \in \{Shifts \setminus \{R,\, C\}\}} \sum_{d' \in Days, d' \leq d} l_s \cdot X_{csd} = \rho_{cd} \quad \forall c \in Crews,\ \forall d \in Days \tag{6}$$

$$\sum_{d' \in Days, d' \leq d} X_{csd} \leq \frac{\rho_{cd}}{WHC} \quad s = \text{'C'},\ \forall c \in Crews,\ \forall d \in Days \tag{7}$$

$$\sum_{d \in Days} X_{csd} = \left\lfloor \frac{\rho_{c(l_d)}}{WHC} \right\rfloor + 1 \qquad s = \text{'C'}, \forall c \in Crews \tag{8}$$

$$\sum_{s \in \{Shifts \setminus \{R,\ C\}\}} \sum_{d' \leq (1+MAX_d),(d+d') \leq l_d} X_{csd} \leq MAX_d \quad \forall c \in Crews,\ \forall d \in Days \tag{9}$$

$$\sum_{c \in Crews} \sum_{s \in \{Shifts \setminus \{R,\ C\}\}} X_{csd} = n_c - \lambda_d \quad \forall d \in Days \tag{10}$$

$$X_{csd} + X_{cs'(d+1)} \leq 1 + \alpha_{css'd} \quad \begin{cases} \forall s, s' \in \{Shifts \setminus \{R,\ C\}\}, s \neq s' \\ \forall c \in Crews, \forall d \in \{Days \setminus \{l_d\}\} \end{cases} \tag{11}$$

$$X_{csd} + X_{cs'(d+1)} \leq 1 + \beta_{css'd} \quad \begin{cases} \forall s, s' \in \{Shifts \setminus \{R,\ C\}\},\ s \neq s',\ with\ t_s \neq t_{s'} \\ \forall c \in Crews, \forall d \in \{Days \setminus \{l_d\}\} \end{cases} \tag{12}$$

$$X_{csd} + X_{cs'd+1} \leq 1 + \gamma'_{css'd} \quad \begin{cases} s \in \{Shifts \setminus \{R,\ C\}\},\ s' = \text{'C'} \\ \forall c \in Crews,\ \forall d \in \{D \setminus \{l_d\}\} \end{cases} \tag{13}$$

$$\sum_{s \in Shifts \setminus \{R,\ C\}} \sum_{d \in Days} X_{csd} = \delta_c \quad \forall c \in Crews \tag{14}$$

$$\sum_{s \in Shifts \setminus \{R,\ C\}} \sum_{d \in Days} l_s \cdot X_{csd} = \theta_c \quad \forall c \in Crews \tag{15}$$

$$\delta_c - \delta_{c'} \leq \phi_{cc'} \quad \forall c, c' \in Crews,\ c \neq c' \tag{16}$$

$$\theta_c - \theta_{c'} \leq \varphi_{cc'} \quad \forall c, c' \in Crews,\ c \neq c' \tag{17}$$

$$\left(\sum_{c \in Crews} X_{csd} - r_s \right) - \left(\sum_{c \in Crews} X_{cs'd} - r_{s'} \right) \leq \psi_{ss'} \quad \begin{cases} \forall s, s' \in \{Shifts \setminus \{R,\ C\}\} \\ \forall d \in Days \end{cases} \tag{18}$$

$$X_{csd},\ \alpha_{css'd},\ \beta_{css'd},\ \gamma_{css'd} \in \{0, 1\} \tag{19}$$

$$\delta_c, \theta_c, \rho_{cd}, \phi_{cc'}, \varphi_{cc'}, \psi_{ss'} \in \mathbb{N} \tag{20}$$

The five terms of the objective function aims at maximizing operating capacity while minimizing the soft constraint violations. The first term (1a) aims at maximizing operating capacity. The weighted sum (1b) assesses the (S1, shift grouping), (S2, same start time) and (S3, compensation assignments) soft constraint violations. The period fairness (S4) soft constraint relates to the number of shift assignment differences and to the working time differences between crews, they are considered using the (1c) term. The preferences of the firefighters (S5) are considered using the (1d) term. The evenly balance of extra daily shifts (S6) is considered using the (1e) term.

The hard constraints **one shift a day** (H1) are enforced by Equation (2). The hard constraints **minimum demands** (H2) are enforced by Equation (3). The hard constraints **forbidden shift successions** (H3) are enforced by Equation (4). The hard constraints **maximum workload** (H4) are enforced by Equation (5). The hard constraints **compensation** (H5) are enforced by Equations (6)-(8). For a crew c and a day d, Equation (6) count ρ_{cd}, the number of worked hours of crew c from the first day of the planning period to day d, and links variables X_{csd} and ρ_{cd}. For a crew c and a day d, Equation (7) forces the number of compensation days ($s =' C'$) being assigned to be less or equal to (ρ_{cd}/WHC) since one compensation day is granted when WHC worked hours are made. For a crew c, all the compensation days must be assigned over the planning horizon (until $d = l_d$), this is enforced by Equation (8). The hard constraints **maximum consecutive working days** (H6) are enforced by Equation (9). For a crew c and a day d, the crew is assigned to at most MAX_d consecutive working shifts (rest and compensation days are not to be considered).

The daily differences between the maximum number of assignable crews (n_c) and those assigned are to be minimized to optimize the overall operating capacity, the λ_d values are assessed by Equation (10).

Consider a crew c, two days d and $d+1$, if the crew is assigned to two different shifts ($s \neq s'$) a **shift grouping** (S1) soft constraint violation occurs and Equation (11) sets $\alpha_{css'd} = 1$. Consider a crew c, two days d and $d+1$, if the crew is assigned to two different shifts ($s \neq s'$) and the start times of these shifts are different ($t_s \neq t_{s'}$) a **same start time** (S2) soft constraint violation occurs and Equation (12) sets $\beta_{css'd} = 1$. Consider a crew c, two day d and $d+1$, if the crew is assigned to a working shift ($s \neq$ 'R') on day d, and if this crew is assigned to a compensation day ($s' =$ 'C') on day $d+1$ a **compensation assignment** (S3) soft constraint violation occurs and Equation (13) sets $\gamma_{css'd} = 1$. Every compensation day assignment will be right after a rest day (constraints of good practice imposed by the institution).

Consider a crew c, Equation (14) counts δ_c the total number of worked shifts over the planning period and Equation (15) counts θ_c the total working time over the planning period. Hence, Equation (16) gives $\phi_{cc'}$ the number of shift assignment differences. Given that $\phi_{cc'} \in \mathbb{N}$, a negative difference involves $\phi_{cc'} = 0$, so for any couple of crews only positive differences are counted. The same rationale applies on Equation (17) for $\varphi_{cc'}$, the number of working time differences. These variables $\phi_{cc'}$ and $\varphi_{cc'}$ are used for the **period fairness** (S4) soft constraint violations assessment.

We recall that **preferences** (S5) soft constraint violations are assessed by Equation (1d).

Consider a day d and two shifts s and s', Equation (18) aims at **evenly balance extra daily shifts** (S6). Minimum demands (H2) are enforced by Equation (3), assigning of extra crews to shifts should be balanced each day within the forest fire period to increase operating capacity.

Equation (19) defines variables X_{csd}, $\alpha_{css'd}$, $\beta_{css'd}$ and $\gamma_{css'd}$ as boolean. Equation (20) defines variables δ_c, θ_c, ρ_{cd}, $\phi_{cc'}$, $\varphi_{cc'}$ and $\psi_{ss'}$ as integers.

4 Adaptive iterative destruction/construction heuristic

We propose an Adaptive Iterative Destruction/Construction Heuristic (AIDCH) to compute solutions of good quality for larger instances of the FFTP. The Algorithm 1 gives the global scheme of the AIDCH proposed approach. We use the adaptive construction approach *BuildFeasibleSchedule()* to build an initial solution which respects the hard constraints. The initial solution complies with *minimum demands* (H2) but there is room for improvement in operating capacity.

Algorithm 1: General structure of AIDCH

```
Input      : An instance of FFTP
Output     : S_best best solution found
Parameters: D_limit limit for diversification degree, n_c number of crews
             n_s number of type of shifts
Variables  : iter number of iterations, MaxIter maximum iteration
             D_max diversification degree, S_cur current solution
iter := 0
MaxIter := n_c
D_max := 3
D_limit := ⌈n_c / n_s⌉
S_cur := BuildFeasibleSchedule()
S_best := S_cur
while iter < MaxIter do
    k := rand(1,D_max)
    AdaptativeDestruction(S_cur,k)  /* adaptive diversification */
    AdaptativeConstruction(S_cur)  /* insert as many crews as possible in S_cur */
    if S_cur > S_best then
        S_best := S_cur
        iter := 0
        D_max := 3
    else
        iter++
        D_max := min(D_max+1,D_limit)
    end
end
```

Provided a feasible solution, at each iteration, a part of the solution is destroyed by removing at random a number k of crews, then it is completed by inserting as many crews as possible in order to increase the operating capacity (while respecting the hard constraints). At each overall iteration at most D_{max} crews are removed ($k \leq D_{max}$). Therefore, we define D_{max} as the degree of

Algorithm 2: Best Insertion Algorithm

```
Input     : S_cur a partial solution
            (α, β, γ, θ, ω, μ) parameter set
Output    : S_best best solution found
Variables : (d,s,c)* best triplet, success boolean
S_best := S_cur  /* store reference solution for BIA */
success := true
while success do
    (d,s,c)* := (∅, ∅, ∅)
    foreach d ∈ Days do
        foreach s ∈ Shifts do
            foreach c ∈ Crews do
                ComputeBIC(d,s,c)
                UpdateBestTriplet (d,s,c)*
            end
        end
    end
    success := Insert(S_cur, (d,s,c)*)  /* if no feasible insertion, Insert returns false */
    /* Comparing S_cur and S_best, all terms of the objective function are assessed */
    if S_cur > S_best then
        S_best := S_cur
    end
end
```

diversification. The D_{max} value is initialized to 3, next incremented after each non-improving overall iteration up to D_{limit}. We set $D_{limit} = \lceil n_c/n_s \rceil$ which represents the average number of crews that can be assigned to shifts. Provided an improvement is found, D_{max} is reset to 3 to entirely explore the neighborhood of the new solution. We perform an adaptive construction procedure to complete the solution. This process is reiterated and it stops when $MaxIter$ overall iterations have been performed without improving the quality of the solution. We set $MaxIter = n_c$. The final result is the best solution found over all iterations.

The proposed AIDCH algorithm makes use of an adaptive diversification mechanism with the aim to escape from local optima. We explore the neighborhood of the new solution as soon as an improvement is found. We explore more distant zones by increasing D_{max} whenever the search is trapped in a local optimum.

The main component of the AIDCH heuristic is the *AdaptativeConstruction(S_{cur})* procedure, an adaptive construction heuristic based on a Best Insertion Algorithm (BIA) shown in Algorithm 2. The BIA algorithm considers a partial solution S_{cur}, and tries to insert as many crews as possible in S_{cur}, one by one. At each iteration, the BIA assesses all feasible insertions that respect the hard constraints and scores them according to a Best Insertion Criterion (BIC). The best insertion is then performed and the quality of S_{cur} is assessed considering all terms of the objective function (1a)-(1e). This process is iterated until no more valid insertion is possible. The algorithm returns the updated S_{cur}, the best solution over all the BIA iterations.

To evaluate the insertion of a crew in the planning (day, shift), we propose to compute the Best Insertion Criterion (BIC) as follows:

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

The aim is to minimize the soft constraints violation whether the insertion is performed. In case a hard constraint is violated (e.g. maximum workload (H4)), the BIC is set to $+\infty$. The criterion is composed of 6 terms, one 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 terms are weighted with parameters α, β, γ, θ, ω and μ in order to control their relative importance.

At each iteration i of the AIDCH heuristic, *AdaptativeConstruction(S_{cur})* works as follows. Four constructive heuristics launch separately BIA with different values of the parameter set $(\alpha, \beta, \gamma, \theta, \omega, \mu)$ on the current solution. During each launch, α, β, γ, θ, ω and μ are chosen randomly in the 6 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, where α_{i-1}, β_{i-1}, γ_{i-1}, θ_{i-1}, ω_{i-1} and μ_{i-1} are the best parameters obtained by the method at previous iteration. All four BIA being applied, the parameter set that produces the best solution is stored to be used in the next iteration.

Finally, the best solution obtained among the four methods is retained as the current solution. This aims at performing parallel independent searches in the solutions space and at choosing the best values of the parameters to better explore the solutions space to speed-up the convergence of the AIDCH algorithm toward a good solution.

5 Computational experiments

In our experiments, our objectives were: (i) to show the adaptive construction impact, by comparing ϕ together with the best parameter set that produces the best solution at previous iteration to compute the next parameter set, versus a fully randomized parameter set; (ii) to show the efficiency of the adaptive destruction, impact of an adaptive D_{max} for perturbations versus a constant one; (iii) to compare performances between the ILP model and the AIDCH approach within a 3600 seconds time limit.

Tests were done using C++ compiled with gcc version 7.5.0, using STL, using a CPLEX 12.10 [8] solver with a single thread and the *MipEmphasis* 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 performance metric

We tested the ILP and AIDCH approaches on a benchmark composed of 4 datasets, each having 7 instances, that we generated using real data of the INFOCA firefighter institution. Datasets have been created to be of increasing difficulty, the firsts of reasonable sizes given that the ILP may face difficulty to get a solution within the time limit. The instances in datasets are ranged according to the number of crews n_c and to the total daily number of working shifts 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.

For each instance, the AIDCH algorithm is run 10 times. We recorded the **R**elative **P**ercentage **E**rror, we defined as $RPE = 100 * (Z_{best} - Z_{max})/Z_{best}$ and the **A**verage **R**elative **P**ercentage **E**rror, we defined as $ARPE = 100 * (Z_{best} - Z_{avg})/Z_{best}$ where Z_{max} is the best result obtained among the ten executions, Z_{avg} is the average result obtained among the ten runs and Z_{best} is the best solution found by the AIDCH approach for the according instance. The ARPE criterion aims at investigating whether the AIDCH is stable over the runs.

To compare the solutions found by the AIDCH approach against the solutions attained by the ILP approach, we define the **R**elative **P**ercentage **G**ap as $RPG = 100*(Z_{ILP} - Z_{max})/Z_{ILP}$ where Z_{ILP} represents the solution value attained, if any, by the ILP approach for an instance.

For our experiments using the ILP, we set w_{oc} to 2, w_{sg} to 1, w_{sst} to 1, w_{ca} to 1 and w_p to 2.

Impact of the adaptive construction mechanism

We first carried out preliminary experiments to choose the best value of ϕ that is necessary to show the impact of the adaptive construction mechanism, because of lack of space those experiments are not reported here. According to these experiments, the parameter value $\phi = 0.1$ provides the best results considering RPE.

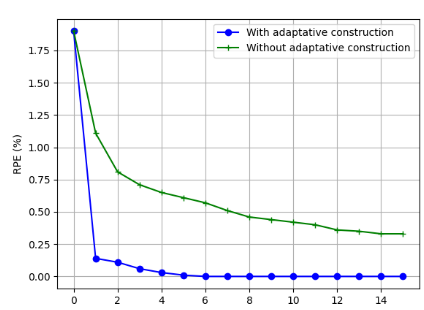

Fig. 1. Adaptive construction impact

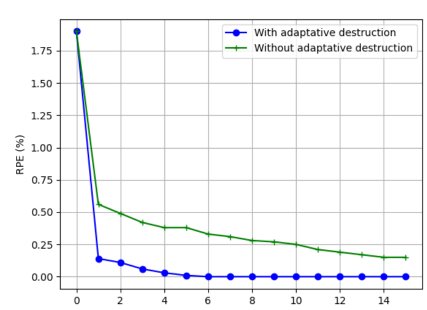

Fig. 2. Adaptive destruction impact

The adaptive construction mechanism aims to guide the search by computing at each time the best trade-off between the different terms of the BIC representing soft constraints violations. To show whether it is efficient, we conducted experiments with the adaptive construction mechanism and without the adaptive construction mechanism. In that latter case, the parameters of BIC are chosen randomly in $[0, 1]$ at each iteration. In these experiments, for each instance, the algorithm is launched and we record the best solution for the first 15000 iterations. We performed these tests using 2 instances chosen at random from each dataset. We report in Figure 1 the average of RPE values computed for the 8 chosen instances against the number of iterations.

As it can be shown in Figure 1, the adaptive construction mechanism permits to converge faster toward good solutions rather than without adaptive construction mechanism.

Impact of the diversification mechanism

To evaluate the effectiveness of the adaptive destruction, we tested a version of AIDCH where the diversification degree D_{max} is set to 3. As aforementioned, we record the best solution for the first 15000 iterations using this fixed value. We proceed in the same way using the adaptive diversification mechanism that makes use of D_{max} to explore the neighborhood of the new solution as soon as an improvement is found and also to explore more distant zones whenever the search is trapped in a local optimum.

Figure 2 shows the average of RPE values recorded against the number of iterations for these two versions. The adaptive diversification mechanism, achieved using the management of D_{max}, permits to converge faster toward good solutions rather than without its use.

Based on these two graphs, we can easily notice that the average of RPE values with the adaptive mechanisms is always below the average of RPE values with the standard perturbation at each iteration, which shows the effectiveness of our proposed technique.

Instance	ILP	t (s)	gap	AIDCH	t (s)	RPG	ARPE	Instance	ILP	t (s)	gap	AIDCH	t (s)	RPG	ARPE
c18r09a	1325	1443	0	1325	341	0	0	c50r22a	ns	-	nc	3765	741	nc	0.43
c18r10a	1359	1409	0	1359	352	0	0	c50r23a	ns	-	nc	3783	754	nc	0.31
c18r10b	1344	1526	0	1344	348	0	0	c50r26a	ns	-	nc	3799	759	nc	0.27
c18r11a	1378	1886	0	1378	372	0	0	c50r28a	ns	-	nc	3823	783	nc	0.58
c18r11b	1420	2786	0	1420	401	0	0	c50r31a	ns	-	nc	3947	849	nc	0.52
c18r12a	1422	2103	0	1422	391	0	0	c50r33a	ns	-	nc	3931	817	nc	0.56
c18r12b	1440	2209	0	1440	413	0	0	c50r35a	ns	-	nc	4097	831	nc	0.34
c30r15a	1767	-	0.74	1758	553	-0.51	0.1	c70r31a	ns	-	nc	4913	943	nc	0.67
c30r16a	1801	-	1.18	1811	561	0.56	0.15	c70r33a	ns	-	nc	4957	954	nc	0.71
c30r17a	1818	-	0.44	1860	582	2.31	0.22	c70r37a	ns	-	nc	5102	995	nc	0.69
c30r18a	1834	-	0.11	1867	593	1.80	0.08	c70r40a	ns	-	nc	5151	1034	nc	0.65
c30r19a	1889	-	0.48	1934	612	2.38	0.13	c70r44a	ns	-	nc	5213	1067	nc	0.71
c30r20a	2144	-	12.9	1947	661	-9.19	0.26	c70r47a	ns	-	nc	5557	1113	nc	0.83
c30r21a	1966	-	2.77	1936	657	-1.53	0.16	c70r50a	ns	-	nc	5401	1158	nc	0.67

Table 1. Performances of ILP and AIDCH approaches

ILP versus AIDCH

Table 1 compares the results obtained by the ILP solver against those obtained by the AIDCH approach. In Table 1, **ns** stands for **n**o **s**olution, **nc** stands for **n**ot **c**alculable, and **-** shows that the 3600 seconds time limit has been attainted. For the sake of compactness, datasets are grouped by two then tabulated side by side. Column *Instance* gives the instance label. The next tree columns, *ILP*, *t (s)* and *gap* show the performances of the ILP. They report the objective function value, the computing time and the gap found by the CPLEX solver. Then, the next four columns, *AIDCH*, *t (s)*, RPG, and *ARPE* show the performances of the AIDCH approach. They report the objective function value, the computing time, the gap between the solutions found by the AIDCH approach and the solution provided by the ILP solver and the average of RPEs over the 10 runs for an instance.

The ILP approach attains optimal solutions for all $n_c = 18$ instances. It faces difficulty for the second dataset having $n_c = 30$, however feasible solutions are obtained within the 3600s time limit.

For the third and the fourth datasets having $n_c = 50$ and $n_c = 70$, the ILP approach fails to find a feasible solution within the time limit.

For the first dataset, the AIDCH approach succeeded in obtaining all the optimal solutions found by the ILP approach. We also notice that all the ARPE values are equal to 0: which means that the AIDCH approach was able to attain the optimal solutions.

For the second dataset, the AIDCH approach attains solutions closed to or better than the solutions obtained by the ILP approach within a 3600s time limit. For four instances the RPG values are between 0.56 and 2.38. For the three other instances, the AIDCH approach obtains better solutions than the ones provided by the ILP approach, with an RPG values from -0.51 up to -9.19. ARPE values are less than 0.26 for all instances which shows the stability of our proposed heuristic approach for this dataset.

For the third and the fourth datasets, the AIDCH approach was able to find solutions in a reasonable time. The ARPE values are less than 0.83, the proposed heuristic behaviour is stable over the last two datasets. Unfortunately, the quality of the solutions found by the AIDCH approach cannot be assessed since the ILP approach fails to provide solutions for these datasets within the one hour time limit.

6 Conclusion and future work

We presented in this paper both an ILP model and a AIDCH heuristic to address the real-worl firefighters timetabling problem (FFTP) of the INFOCA institution. The proposed approaches were tested over four datasets with different sizes of increasing difficulty that we generated using real data from INFOCA. The ILP approach obtained optimal or near optimal solutions for the first two datasets, but it faced difficulty in obtaining feasible solutions for the larger instances of the two other datasets. The AIDCH approach obtained good solutions for all the instances of the first two datasets, those are either optimal or closed to the ones obtained by the ILP approach. The proposed heuristic approach was able to find feasible solutions for the larger instances within a reasonable computation time. Future works aim at investigating a metaheuristic solution approach to improve the quality of the solutions obtained over the datasets and aim at reducing the computation time. We also plan to obtain lower bounds for the larger instances for comparison purposes.

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 program 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).

References

1. Aggarwal, S. C. (1982). A focussed review of scheduling in services. European Journal of Operational Research, 9(2), 114-121.
2. Barnhart, C., Cohn, A. M., Johnson, E. L., Klabjan, D., Nemhauser, G. L., Vance, P. H. (2003). Airline crew scheduling. In Handbook of transportation science (pp. 517-560). Springer, Boston, MA.
3. Ben-Said, A., El-Hajj, R., Moukrim, A. (2016). An adaptive heuristic for the Capacitated TeamOrienteering Problem. In 8th IFAC Conference on Manufacturing Modelling, Management and Control (pp 1662-1666).
4. Burke, E. K., De Causmaecker, P., Berghe, G. V., Van Landeghem, H. (2004). The state of the art of nurse rostering. Journal of scheduling, 7(6), 441-499.
5. Edie, L. C. (1954). Traffic delays at toll booths. Journal of the operations research society of America, 2(2), 107-138.
6. Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D. (2004). An annotated bibliography of personnel scheduling and rostering. Annals of Operations Research, 127(1), 21-144.
7. 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.
8. IBM, CPLEX, User's Manual, https://www.ibm.com, (2020)
9. Tien, J. M., Kamiyama, A. (1982). On manpower scheduling algorithms. SIAM review, 24(3), 275-287.
10. 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.