



Application of decomposition techniques in a wildfire suppression optimization model

Jorge Rodríguez-Veiga^{1,3} · David R. Penas^{3,4} · Ángel M. González-Rueda² · María José Ginzo-Villamayor³

Received: 19 July 2022 / Revised: 1 December 2022 / Accepted: 15 December 2022
© The Author(s) 2023

Abstract

Resource assignment and scheduling models provides an automatic and fast decision support system for wildfire suppression logistics. However, this process generates challenging optimization problems in many real-world cases, and the computational time becomes a critical issue, especially in realistic-size instances. Thus, to overcome that limitation, this work studies and applies a set of decomposition techniques such as augmented Lagrangian, branch and price, and Benders decomposition's to a wildfire suppression model. Moreover, a reformulation strategy, inspired by Benders' decomposition, is also introduced and demonstrated. Finally, a numerical study comparing the behavior of the proposals using different problem sizes is conducted.

Keywords Integer programming · Assignment problems · Wildfire management · Decomposition techniques · Benders decomposition

✉ María José Ginzo-Villamayor
mariajose.ginzo@usc.es

Jorge Rodríguez-Veiga
jorge.rodriguez.veiga@gmail.com

David R. Penas
david.rodriguez.penas@csic.es

Ángel M. González-Rueda
angel.manuel.rueda@udc.es

¹ Technological Institute of Industrial Mathematics (ITMATI), Santiago de Compostela, Spain

² Department of Mathematics, Modes Research Group and CITIC, University of A Coruña, A Coruña, Spain

³ Modestya Research Group, Department of Statistics, Mathematical Analysis and Optimization, University of Santiago de Compostela, Santiago de Compostela, Spain

⁴ Computational Biology Lab, MBG-CSIC, Spanish National Research Council, Pontevedra, Spain

1 Introduction

Many critical problems in disaster management and logistics can be formulated and solved using the Operations Research (OR) framework (Caunhye et al. 2012). Wildfires are a type of catastrophe with a high impact on the current world, in humanitarian, economic, and above all, ecological terms (European Commission 2020). As their frequency and magnitude are growing at an alarming rate, it is necessary to develop efficient methods to improve the prevention, detection, and planning in logistics related to wildfire suppression.

In Galicia (Northwest of Spain), forest fires are a severe problem to be faced every year, caused by deforestation, arsonists, or the removal of local flora and fauna. With just a surface area of 29,574 km², this region had more than 3550 fires per year from 2011 and 2015, being 42,392.17 hectares burned in 2011 alone. Typically, the wildfires in Galicia are active between 1 and 7 days, requiring more than 70 brigades, 50 engines, and 20 aircraft on the largest fires. The Regional Government of Galicia has to periodically manage (especially in summer) a large amount of wildfire fighting resources (Xunta de Galicia 2017). For instance, they had 7000 people, 360 motor pumps, and 30 aerial resources (25 were helicopters) in the 2017 campaign. Consequently, finding a good resource assignment and scheduling avoids its over-allocation, which is very important since they are limited.

Formulating the problem above as a scheduling optimization model will help reduce the impact of wildfires through optimal resource planning. Although several research topics in the optimization of forest fire management have been proposed in recent years (Minas et al. 2012; Miller and Ager 2013; Martell 2015), analyzing the wildfire extinction from the perspective of cost minimization is a well-studied topic throughout history, already introduced in works such as Headley (1916) and Sparhawk (1925). An overall theoretical framework that pursues management of resources based on minimizing costs was C+NVC (Cost plus Net Value Change), proposed in Gorte and Gorte (1979). This model combines the goal of minimizing the cost of resources and the costs generated by burned land, loss of materials or regeneration tasks. More recently, Donovan and Rideout (2003) complemented the C+NVC scheme using a deterministic programming model, which can establish the optimal planning regarding the number and type of resources needed to extinguish a forest fire from its initial detection.

Inspired by the work above and motivated by the peculiarities of the Galician fire extinguishing system, Rodríguez-Veiga et al. (2018) proposed a Mixed-Integer Linear Programming (MILP) model to plan resources involved in a single wildfire extinction. The proposed model was thought to solve problems in the near future, where the uncertainty of the wildfire evolution is lower, for example, in a timeline of 8 h. Thus, this novel model included new features to manage resources such as rest policies or initial conditions, and it could easily be applied to other similar scheduling problems.

On the other hand, in the real world, the resource planning associated with wildfire suppression is managed by a resource coordinator who oversees the schedules of all logistic operations. Mathematical optimization models help to perform these tasks. They provide high-quality solutions to minimize the impacts caused by a wildfire. It implies that time is a critical issue. Conducting optimization that takes hours to reach

optimality does not make sense when a wildfire is occurring. The resource coordinator must react quickly and cannot afford to wait a long time.

In this way, mathematical decomposition methods (Conejo et al. 2006) can offer a good alternative to simplify the models and reduce the computational time. These techniques split a problem into more manageable subproblems based on their complicating variables or constraints (nondecomposable elements). However, solving these subproblems is not an easy task, and it is necessary to design tailor-made algorithms.

Therefore, this manuscript studies several reformulations, based on decomposition techniques in most cases, to improve the MILP model presented in Rodríguez-Veiga et al. (2018). In addition, we compare our proposals with the solving MILP problem above in terms of solution optimality and computation time. Concerning our contribution, we improve solving problems without adding untractable complexity to the mathematical model or optimization process. Besides, our reformulation procedures can easily export to other optimization models related to optimal resource planning.

The organization of this document is as follows. Section 1.1 covers the related work while Sect. 1.2 presents a brief revision of the optimization model to be addressed: a logistic scheduling model for wildfire suppression. Section 2 describes different ways to apply decomposition strategies to the previous model, and Sect. 3 proposes a customized method based on the Benders decomposition. In Sect. 4, the performance of our proposals is evaluated using simulated data of the wildfire model and solving instances of different sizes. Finally, Sect. 5 summarizes the main conclusions of our study.

1.1 Related work

Good planning in extinction resources generally reduces the costs and damages associated with forest fires. Accordingly, one of the most often topics in wildfire literature is the scheduling problem. Such is the case presented in Wei et al. (2011), where a mixed-integer optimization model considers multiple fires, in addition to spatial and time data. Likewise, Ntaimo et al. (2012) proposed a stochastic integer programming (IP) model to the initial attack in as many fires as possible. In this manner, the authors figure out the best resource allocation in each firefight base to contain as quickly as possible several forest fires.

However, in our particular case, Rodríguez-Veiga et al. (2018) proposed a different approach to fire extinction. Our model considers all scheduling management (breaks, use policies) for all the resources involved just in a large-scale wildfire via a deterministic mixed-integer model. In this way, it is in charge of assigning the resources throughout the fire.

As stated above, applying decomposition techniques can improve the solving of optimization problems (Conejo et al. 2006). There are many examples of these techniques used in scheduling problems that contemplate recurring elements in the context of a wildfire.

Works as Cordeau et al. (2001) or Papadakos (2009) have combined Benders decomposition and column generation methods to deal with optimal scheduling in air traffic. Although this work is interesting due to similarities with our model (fleet assign-

ment assignments over time), their authors considered non-existing features under our perspective regarding maintenance routing and crew pairing.

Rios and Ross (2010) presented a decomposition scheme based Dantzig-Wolfe method to ease huge traffic flow scheduling problems. It involves both planning and routing decisions, contrary to our model where we just considered optimal resource scheduling. Thus, this strategy obtained one flight per subproblem, being particularly suitable for parallel computing, and solving instances in a planning time of up to 3 h. In comparison, our proposal needs to handle instances with a higher time threshold, though with fewer resources to allocate.

In addition, Romanski and Hentenryck (2016) considered prescriptive evacuation planning and route design for a region threatened by a natural disaster, such as a flood, forest fire, or a hurricane. They proposed a Benders decomposition that generalizes the two-step approach proposed in previous works for converging evacuation plans. On the contrary to our model, they need to deal with flow scheduling problems. Nevertheless, this work is a good example of applying a decomposition strategy to improve a model related to disaster management.

As far as we know, there is no work dedicated to implementing mathematical decomposition techniques in optimization models in the context of resource planning in wildfires. We hope to contribute to simplifying the application of these effective strategies in different forestry optimization problems.

1.2 Problem statement: a wildfire suppression model

We begin from the wildfire extinguishment model presented in Rodríguez-Veiga et al. (2018) In the same way as Donovan and Rideout (2003), we made the following assumptions:

- Some characteristics of firefighting resources are predefined. Such is the case of the fireline production rate, arrival time, or operating costs.
- A resource will start a rest when it reaches the maximum working time. Also, resources can leave a specific wildfire just once time.
- The extinction time was discretized in 10 min periods, considered enough time for a resource to perform a task. This time is the maximum common divisor between the rest periods (40 min), activity periods (120 min) and estimated time to travel to a rest point (10 min) according to the Spanish regulation.
- The model is designed to work over a single wildfire with a maximum planning horizon of 10 h, where predictions of its growth use smaller time intervals (e.g., 10 min). Currently, simulation tools, such as *FARSITE* (Finney 1998) or *Xeocode2* (Xunta de Galicia 2017), can create a predictive 10 h model through GIS information about the wildfire evolution. Thus, an advantage in our model is that it can easily rerun and adapt to changes in the forecast, following a rolling horizon approach (Sethi and Sorger 1991).
- There is no direct interaction between fire perimeter and fireline construction. This assumption is valid due to being considered a pessimistic approach for the wildfire contention.

Table 1 Sets and parameters of the problem

Sets	Definition
i, I	Index and set of resources
g, \mathcal{G}	Index and set of different resources groups
t, \mathcal{T}	Index and set of time periods with wildfire behaviour information $\{1, \dots, m\}$, where m represents the number of periods
i, \mathcal{I}_g	Index and set of resources associated with group g
$t, \mathcal{T}_{t_1}^{t_2}$	Index and set of time periods between the natural numbers t_1 and t_2 , $\{\max\{1, t_1\}, \dots, \min\{m, t_2\}\}$. If the sub-index or super-index is omitted, it means that $t_1 = 1$ and $t_2 = m$, respectively. When t_1 is greater than t_2 , we will consider it an empty set
Parameters	Definition
C_i	Cost per usage period of resource $i \in \mathcal{I}$
P_i	Fixed cost for resource $i \in \mathcal{I}$ selection
NVC_t	Increase in the costs of the wildfire in period $t \in \mathcal{T}$
PER_t	Increment of the wildfire perimeter in period $t \in \mathcal{T}$
PR_{it}	Performance of resource $i \in \mathcal{I}$ in period $t \in \mathcal{T}$
ITW_i	Value 1 indicates resource $i \in \mathcal{I}$ is currently working in this wildfire (0 otherwise)
IOW_i	Value 1 indicates resource $i \in \mathcal{I}$ is currently working in other wildfire (0 otherwise)
CWP_i	Number of periods used since the last final rest of $i \in \mathcal{I}$
CRP_i	Number of rest periods used by resource $i \in \mathcal{I}$
CUP_i	Number of usage periods in the day of resource $i \in \mathcal{I}$
A_i	Number of periods needed by $i \in \mathcal{I}$ to arrive to the wildfire
WP_i	Maximum allowed number of periods without breaks for $i \in \mathcal{I}$
TRP_i	Number of periods needed by resource $i \in \mathcal{I}$ to go from the resting point to the fire
RP_i	Number of rest periods that resource $i \in \mathcal{I}$ must do on a break before starting to work again
UP_i	Maximum number of allowed usage periods in a day for $i \in \mathcal{I}$
$nMin_{gt}$	Minimum number of resources of group $g \in \mathcal{G}$ working on the wildfire in the same period $t \in \mathcal{T}$
$nMax_{gt}$	Maximum number of resources of group $g \in \mathcal{G}$ working on the wildfire in the same period $t \in \mathcal{T}$
M'	Positive constant that penalizes the breach of the minimum number of resources in each period
M	Positive sufficiently large constant to establish wildfire containment. A suitable value is $M := \sum_{t \in \mathcal{T}} PER_t$

Table 2 Variables of the problem

Variables	Definition
$s_{it} \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is starting to be used in period $t \in \mathcal{T}$
$tr_{it} \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is travelling in period $t \in \mathcal{T}$
$r_{it} \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is resting in period $t \in \mathcal{T}$
$er_{it} \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is ending a break in period $t \in \mathcal{T}$
$e_{it} \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is ending its work in period $t \in \mathcal{T}$
$y_t \in \{0, 1\}$	It takes the value 1 if the fire is not contained in period $t \in \{0\} \cup \mathcal{T}$
$\mu_{gt} \in \mathbb{N}$	It indicates the number of missing resources of group $g \in \mathcal{G}$ to reach the corresponding minimum in period $t \in \mathcal{T}$
Aux Vars	Definition
u_{it}	Resource $i \in \mathcal{I}$ has a task assigned in period $t \in \mathcal{T}$ $u_{it} := \sum_{t' \in \mathcal{T}^t} s_{it'} - \sum_{t' \in \mathcal{T}^{t-1}} e_{it'}$
w_{it}	Resource $i \in \mathcal{I}$ works fighting the wildfire in period $t \in \mathcal{T}$ $w_{it} := u_{it} - r_{it} - tr_{it}$
z_i	Resource $i \in \mathcal{I}$ is selected to fight the wildfire $z_i := \sum_{t \in \mathcal{T}} e_{it}$
cr_i	Number of periods since the last ending break period of resource $i \in \mathcal{I}$ in period $t \in \mathcal{T}$ $cr_{it} := \begin{cases} \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} \\ - \sum_{t' \in \mathcal{T}^t} (t - t') \cdot e_{it'} - \sum_{t' \in \mathcal{T}^t} r_{it'} & \text{if } ITW_i = 0 \\ & \text{and } IOW_i = 0 \\ - \sum_{t' \in \mathcal{T}^t} WP_i \cdot er_{it'} \\ (t + CWP_i - CRP_i) \cdot s_{i1} \\ + \sum_{t' \in \mathcal{T}_2^t} (t + 1 - t' + WP_i) \cdot s_{it'} \\ - \sum_{t' \in \mathcal{T}^t} (t - t') \cdot e_{it'} - \sum_{t' \in \mathcal{T}^t} r_{it'} & \text{in the other case} \\ - \sum_{t' \in \mathcal{T}^t} WP_i \cdot er_{it'} \end{cases}$

A short description of this model is shown in the following paragraphs. Table 1 shows a definition of the different sets and parameters used in the model, and Table 2 describes the decision and auxiliary variables. The objective function and the constraints of the model are formulated as follows:

$$\text{minimize} \quad \sum_{i \in \mathcal{I}, t \in \mathcal{T}} C_i \cdot u_{it} + \sum_{i \in \mathcal{I}} P_i \cdot z_i + \sum_{t \in \mathcal{T}} NVC_t \cdot y_{t-1} + \sum_{g \in \mathcal{G}, t \in \mathcal{T}} M' \cdot \mu_{gt} \quad (1)$$

subject to

$$\sum_{t \in \mathcal{T}} PER_t \cdot y_{t-1} \leq \sum_{i \in \mathcal{I}, t \in \mathcal{T}} PR_{it} \cdot w_{it}, \tag{2}$$

$$M \cdot y_t \geq \sum_{t' \in \mathcal{T}^t} PER_{t'} \cdot y_{t-1} - \sum_{i \in \mathcal{I}, t' \in \mathcal{T}^t} PR_{it'} \cdot w_{it'} \quad \forall t \in \mathcal{T}, \tag{3}$$

$$A_i \cdot w_{it} \leq \sum_{t' \in \mathcal{T}^t} tr_{it'}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{4}$$

$$s_{i1} + \sum_{t \in \mathcal{T}_2} (m + 1) \cdot s_{it} \leq m \cdot z_i, \quad \forall i \in \mathcal{I} : ITW_i = 1, \tag{5}$$

$$\sum_{t \in \mathcal{T}} s_{it} \leq z_i, \quad \forall i \in \mathcal{I} : ITW_i = 0, \tag{6}$$

$$\sum_{t' \in \mathcal{T}_{t-TRP_i+1}^t} tr_{it'} \geq TRP_i \cdot e_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{7}$$

$$0 \leq cr_{it} \leq WP_i, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{8}$$

$$r_{it} \leq \sum_{t' \in \mathcal{T}_i^{t+RP_i-1}} er_{it'}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{9}$$

$$\sum_{t' \in \mathcal{T}_{t-RP_i+1}^t} r_{it'} \geq RP_i \cdot er_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_{RP_i}, \tag{10}$$

$$CRP_i \cdot s_{i1} + \sum_{t' \in \mathcal{T}^t} r_{it'} \geq RP_i \cdot er_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}^{RP_i-1}, \tag{11}$$

$$\sum_{t' \in \mathcal{T}_{t-TRP_i}^{t+TRP_i}} (r_{it'} + tr_{it'}) \geq \sum_{t' \in \mathcal{T}_{t-TRP_i}^{t+TRP_i}} r_{it'}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{12}$$

$$\sum_{t \in \mathcal{T}} u_{it} \leq UP_i - CUP_i, \quad \forall i \in \mathcal{I}, \tag{13}$$

$$\sum_{i \in \mathcal{I}_g} w_{it} \geq nMin_{gt} \cdot y_{t-1} - \mu_{gt}, \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{14}$$

$$\sum_{i \in \mathcal{I}_g} w_{it} \leq nMax_{gt} \cdot y_{t-1}, \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{15}$$

$$\sum_{t \in \mathcal{T}} t \cdot e_{it} \geq \sum_{t \in \mathcal{T}} t \cdot s_{it}, \quad \forall i \in \mathcal{I}, \tag{16}$$

$$\sum_{t \in \mathcal{T}} e_{it} \leq 1, \quad \forall i \in \mathcal{I}, \tag{17}$$

$$r_{it} + tr_{it} \leq u_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{18}$$

$$\sum_{t \in \mathcal{T}} w_{it} \geq z_i, \quad \forall i \in \mathcal{I}. \tag{19}$$

The objective function (1) minimizes the sum of the costs involved in the wildfire contention: the first term represents the variable costs for the use of the selected resources, the second term represents their selection costs, the third term represents the costs associated with the hectares of affected land, and the last term is included to penalize the breach of the minimum number of group resources in each period.

Regarding the constraints, (2) imposes that the total perimeter covered by the resources must be greater than the wildfire perimeter in the contained period, (3) establishes the conditions under which the wildfire can be contained or not in each period, (4)–(6) are related to the coherent beginning of resource activities, and (7) indicates when a resource ends its job with enough time to return to its operational base. Moreover, (8)–(13) correspond to making task assignment feasible according to legal regulations, and (14)–(15) determine the minimum and the maximum number of resources in the wildfire while the fire is not contained, respectively. Finally, (16)–(19) determine logical conditions that the variables must satisfy.

2 Applying decomposition methods to the problem

The following subsections explain in detail how to apply three popular decomposition techniques to the wildfire suppression model described above, hereinafter called the Original Problem.

2.1 Augmented Lagrangian decomposition

The Augmented Lagrangian (AL) is a framework of penalty methods extensively studied in recent decades (Hamdi and Mishra 2011). It consists of seeking to simplify the Original Problem, moving the complicating constraints to the objective function, and multiplying them by Lagrange multipliers and penalty constants. Furthermore, the iterative calibration of these penalties allows good-quality feasible solutions to be reached.

An AL problem can be decomposed for given values of these Lagrange multipliers, and this procedure is known as Augmented Lagrangian decomposition (ALD). We adjusted this procedure, creating a set of different subproblems. In this way, each subproblem maps a particular resource (aircraft, brigade, or machine) and considers the other resources' values as constants. Consequently, there is the same number of subproblems as different resources in the model. The details of the implementation can be seen in "Appendix A".

2.2 Branch and price decomposition

Branch and price (BPD) (Barnhart et al. 1998; Vanderbeck and Wolsey 1996; Vanderbeck and Savelsbergh 2006) is a method specifically conceived to handle large-scale integer programming problems. It is based on the well-known branch and bound algorithm: it explores feasible solutions, solving in each node a linear programming (LP) relaxation of the problem and branching according to this result. In the case of branch and price, only a subset of columns, usually related to basic variables, remain in the LP relaxation. The other columns are omitted because they are too many and typically have an associated decision variable equal to zero in the optimal solution. Thus, when this method is used in conjunction with Dantzig-Wolfe Decomposition (DWD), we can refer to it as a whole as Branch and Price Decomposition (BPD), being a good alternative to decrease the complexity of large-scale MILP problems.

The DWD is a column generation procedure for continuous problems that instead of considering all the variables of the problem at once, in each step, it solves a master problem containing only some active columns. Furthermore, some subproblems are solved iteratively to determine the columns that must be added to the master problem to improve the objective function.

The BPD is suitable when the problem presents some complicating constraints that prevent a distributed solution into several subproblems defining more tractable combinatorial structure, specially designed to handle integer programming. In this way, the BPD algorithm applies the DWD procedure for each node in the branch and bound tree.

In our case, we want to decompose the Original Problem into several subproblems depending only on resource $i \in \mathcal{I}$, so the resulting subproblems will be more tractable. It is easy to identify that the complicating constraints avoiding this decomposable structure are the following: (2), (3), (14) and (15). Therefore, these constraints must be relaxed and appear penalized in the objective function of the subproblems.

BPD Master Problem

Suppose that we know p solutions of the problem that we denote as SOL_k , for all $k = 1, \dots, p$. Then,

$$SOL_s = (s_{11}^k, \dots, s_{nm}^k, tr_{11}^k, \dots, tr_{nm}^k, r_{11}^k, \dots, r_{nm}^k, er_{11}^k, \dots, er_{nm}^k, e_{11}^k, \dots, e_{nm}^k, y_1^k, \dots, y_m^k, \mu_{11}^k, \dots, \mu_{gm}^k).$$

The BPD master problem can be formulated as follows:

$$\text{minimize } \sum_{k=1}^p f^{(k)} \cdot \alpha_k$$

subject to

$$\sum_{k=1}^p h^{(k)(2)} \cdot \alpha_k \leq 0, \tag{20}$$

$$\sum_{k=1}^p h_t^{(k)(3)} \cdot \alpha_k \leq 0, \quad \forall t \in \mathcal{T}, \tag{21}$$

$$\sum_{k=1}^p h_{gt}^{(k)(14)} \cdot \alpha_k \leq 0, \quad \forall t \in \mathcal{T}, \forall g \in \mathcal{G}, \tag{22}$$

$$\sum_{k=1}^p h_{gt}^{(k)(15)} \cdot \alpha_k \leq 0, \quad \forall t \in \mathcal{T}, \forall g \in \mathcal{G}, \tag{23}$$

$$\sum_{k=1}^p \alpha_k = 1, \tag{24}$$

$$\alpha_k \in \{0, 1\}, \quad \forall k \in \{1, \dots, p\}, \tag{25}$$

where $f^{(k)}$ is the value of the objective function (1) associated with SOL_k ,

$$f^{(k)} = \sum_{i \in \mathcal{I}, t \in \mathcal{T}} C_i \cdot u_{it}^k + \sum_{i \in \mathcal{I}} P_i \cdot z_i^k + \sum_{t \in \mathcal{T}} NVC_t \cdot y_{t-1}^k + \sum_{g \in \mathcal{G}, t \in \mathcal{T}} M' \cdot \mu_{gt}^k,$$

α_k is the binary variable determining whether solution k is selected, and $h^{(k)(2)}, h_t^{(k)(3)}, h_{gt}^{(k)(14)}$ and $h_{gt}^{(k)(15)}$ are the values of the complicating constraints (2), (3), (14) and (15), associated with solution SOL_s , respectively,

$$h^{(k)(2)} = \sum_{t \in \mathcal{T}} PER_t \cdot y_{t-1}^k - \sum_{i \in \mathcal{I}, t \in \mathcal{T}} PR_{it} \cdot w_{it}^k,$$

$$h_t^{(k)(3)} = \sum_{t' \in \mathcal{T}^t} PER_{t'} \cdot y_{t-1}^k - \sum_{i \in \mathcal{I}, t' \in \mathcal{T}^t} PR_{it'} \cdot w_{it'}^k - M \cdot y_t^k,$$

$$h_{gt}^{(k)(14)} = nMin_{gt} \cdot y_{t-1}^k - \mu_{gt}^k - \sum_{i \in \mathcal{I}_g} w_{it}^k,$$

$$h_{gt}^{(k)(15)} = \sum_{i \in \mathcal{I}_g} w_{it}^k - nMax_{gt} \cdot y_{t-1}^k.$$

BPD Subproblem

Suppose that we solve the LP relaxation of the BPD master problem,¹ and let us denote $\lambda^{(2)}, \lambda_t^{(3)}, \lambda_{gt}^{(14)}, \lambda_{gt}^{(15)}$ and σ as the dual solution values associated with constraints (20), (21), (22), (23) and (24), respectively.

¹ It corresponds to relaxing the integrality conditions over the α_k variables.

Then, the formulation of the (aggregated) BPD subproblem can be expressed as:

$$\begin{aligned}
 \text{minimize} \quad & \sum_{i \in \mathcal{I}, t \in \mathcal{T}} C_i \cdot \mathbf{u}_{it} + \sum_{i \in \mathcal{I}} P_i \cdot \mathbf{z}_i + \sum_{t \in \mathcal{T}} NVC_t \cdot \mathbf{y}_{t-1} + \sum_{g \in \mathcal{G}, t \in \mathcal{T}} M' \cdot \boldsymbol{\mu}_{gt} \\
 & - \lambda^{(2)} \left(\sum_{t \in \mathcal{T}} PER_t \cdot \mathbf{y}_{t-1} - \sum_{i \in \mathcal{I}, t \in \mathcal{T}} PR_{it} \cdot \mathbf{w}_{it} \right) - \sum_{i \in \mathcal{I}} \\
 & - \lambda_t^{(3)} \left(\sum_{t' \in \mathcal{T}^t} PER_{t'} \cdot \mathbf{y}_{t-1} - \sum_{i \in \mathcal{I}, t' \in \mathcal{T}^t} PR_{it'} \cdot \mathbf{w}_{it'} - M \cdot \mathbf{y}_t \right) \\
 & - \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \lambda_{gt}^{(14)} (nMin_{gt} \cdot \mathbf{y}_{t-1} - \boldsymbol{\mu}_{gt} - \sum_{i \in \mathcal{I}_g} \mathbf{w}_{it}) \\
 & - \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \lambda_{gt}^{(15)} \left(\sum_{i \in \mathcal{I}_g} \mathbf{w}_{it} - nMax_{gt} \cdot \mathbf{y}_{t-1} \right) - \sigma
 \end{aligned}$$

subject to

$$(4) - (13), (16) - (19).$$

First, it is important to note that this BPD subproblem is decomposable by resource $i \in \mathcal{I}$, obtaining $|\mathcal{I}|$ subproblems that can be solved independently. Once the subproblem is solved, depending on the value of the objective function associated with this new solution, one can determine whether to include this tentative basic new solution (associated with the variable α that represents the weight given to each solution added to the master problem) to the BPD master problem. If the solution improves the objective function, then this new solution is added to the BPD master problem and solved again, giving new dual value solutions to update the BPD subproblem. Thus, this procedure is repeated until no more columns can be added. Note that the scheme we have just described must be applied at every node of a branch and bound procedure.

2.3 Benders decomposition

Contrary to the previous methods, Benders decomposition (BD) (Benders 1962; Rahmaniani et al. 2017) aims to divide the optimization problem based on their complicating variables: those variables which, when they are fixed, the remaining problem will be relatively easy to optimize.

Applying BD to our wildfire suppression model, we split the problem into a master problem and a subproblem. On the one hand, the master problem selects the resources which are able to contain the wildfire (complicating variables). On the other hand, the subproblem obtains the best feasible solutions according to wildfire rest policies and the resource’s state at every moment.

In this way, we have adapted the BD method by adding a set of transformations to the Original Problem based on the following premise: if we know when a resource starts to work in the wildfire, we can determine its future states over time horizon. That is, we can recalculate when this resource is flying towards the fire, in which periods it will work in the fire-front, or during which periods it will need to be resting according to break policies. Consequently, to take advantage of this idea, it is necessary to modify

the Original Problem to facilitate separability. This new reformulation will consider how resources must perform rest and travel periods from the start period until the last period (period m). Furthermore, this information is kept in mind to establish when resources must rest, travel, and work while they are in use, that is, until variable e_{it} takes the value 1.

Reformulated Problem

The transformations performed in the Original Problem can be classified into two groups. First, three new kind of variables have been added associated with the traveling times of the resources. Thus, when a resource travels, it could be moving to the wildfire front, going to rest in the base, or moving due to the end of work (see Table 3). Second, as stated before, suppose we know when a resource starts to work in the wildfire, we can determine how it should act over the timeline. To ease the handle of these periods to be calculated, we have introduced a new group of variables, denoted by a hat over them: traveling associated with rest periods (\hat{tr}_{it}^r), resting (\hat{r}_{it}), ending break (\hat{e}_{it}) and ending (\hat{e}_{it}). Auxiliary variables \hat{u}_{it} , \hat{w}_{it} and \hat{c}_{it} are also defined using the expression of the Original Problem but replacing tr_{it} , r_{it} , er_{it} and e_{it} with \hat{tr}_{it}^r , \hat{r}_{it} , \hat{e}_{it} and \hat{e}_{it} , respectively.

Table 3 Definition of travel variables

Variables	Definition
$tr_{it}^s \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is travelling to go to the wildfire in period $t \in \mathcal{T}$ (travelling associated with start period)
$tr_{it}^r \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is travelling to perform a rest period or to return to the wildfire in period $t \in \mathcal{T}$ (travelling associated with breaks)
$tr_{it}^e \in \{0, 1\}$	It takes the value 1 if resource $i \in \mathcal{I}$ is travelling to leave the wildfire in period $t \in \mathcal{T}$ (travelling associated with end period)

As a result of applying these transformations, the reformulated problem can be expressed as:

minimize (1)

subject to

(2), (3), (5), (6), (13), (14), (15), (16), (17), (18), (19),

$$A_i \cdot w_{it} \leq \sum_{t' \in \mathcal{T}^t} tr_{it'}^s, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{r4}$$

$$\sum_{t' \in \mathcal{T}_{t-TRP_i+1}^t} tr_{it'}^e \geq TRP_i \cdot e_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{r7}$$

$$0 \leq \hat{c}_{it} \leq W P_i, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{r8}$$

$$\hat{r}_{it} \leq \sum_{t' \in \mathcal{T}_t^{t+RP_i-1}} \hat{e}_{it'}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{r9}$$

$$\sum_{t' \in \mathcal{T}_{t-RP_i+1}^t} \hat{r}_{it'} \geq RP_i \cdot \hat{e}r_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_{RP_i}, \quad (r10)$$

$$CRP_i \cdot s_{i1} + \sum_{t' \in \mathcal{T}^t} \hat{r}_{it'} \geq RP_i \cdot \hat{e}r_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}^{RP_i-1}, \quad (r11)$$

$$\sum_{t' \in \mathcal{T}_{t-TRP_i}^{t+TRP_i}} (\hat{r}_{it'} + \hat{t}r_{it'}^r) \geq \sum_{t' \in \mathcal{T}_{t-TRP_i}^{t+TRP_i}} \hat{r}_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (r12)$$

$$\hat{r}_{it} + \hat{t}r_{it}^r \leq \hat{u}_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}. \quad (r18)$$

Moreover, the following constraints are introduced to establish a relation between the original variables and the new variables:

$$u_{it} \leq \hat{u}_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (26)$$

$$r_{it} = u_{it} \cdot \hat{r}_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (27)$$

$$er_{it} = u_{it} \cdot \hat{e}r_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (28)$$

$$w_{it} = u_{it} \cdot \hat{w}_{it} \cdot (1 - \max\{tr_{it}^s, tr_{it}^e\}), \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (29)$$

$$tr_{it}^r = u_{it} \cdot \hat{t}r_{it}^r, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (30)$$

$$tr_{it} = \max\{tr_{it}^s, tr_{it}^e, tr_{it}^r\}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}. \quad (31)$$

In order to explain the reformulated problem, we illustrate the performed transformations with an example.

Example 1 Let us consider a problem instance with a single aerial resource ($\mathcal{I} = \{1\}$) and a time horizon of 9 periods ($\mathcal{T} = \{1, \dots, 9\}$). Suppose that the resource starts without initial conditions ($CWP_1 = CRP_1 = CUP_1 = ITW_1 = IOW_1 = 0$) and it requires 1 period to reach the wildfire from its starting location ($A_1 = 1$). The resource performance will be 1 for all periods ($PR_{1t} = 1$ for all $t \in \mathcal{T}$). In addition, the maximum number of allowed periods without breaks is 4 ($WP_1 = 4$), the duration of the break is limited to 1 ($RP_1 = 1$), and the resource needs 1 travel period ($TRP_1 = 1$) to move between fire-front and the rest location.

In this context, suppose that the wildfire has an initial perimeter of 2 km and it grows 0.1 km per period ($PER_1 = 2$ and $PER_t = 0.1$ for all $t \in \{2, \dots, 9\}$).

Figure 1 shows the solution (active variables for each period) of the given instance to represent the main idea of the reformulated problem. The figure illustrates how the new variables split the model into two parts to satisfy the rest policy.

- The new variables (those over the edges) denoted with the hat represent how the resource must perform the rest periods. Furthermore, it also considers travel due to rest periods.

- The original variables (those below the edges) represent how the resource works in the wildfire to contain it, ensuring that the travel and rest periods established by the new variables must be performed.

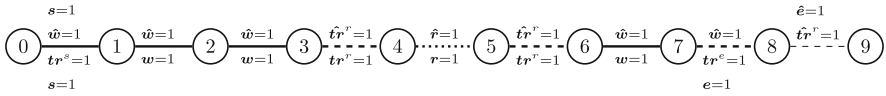


Fig. 1 Illustration of the relation between original variables and the new ones

Figure 1 shows how the resource starts to be used in the wildfire in the first period. At this moment, although variable \hat{w} indicates that the resource could work, constraints (r4) and (29) force the resource to perform a starting flight before beginning to work. In periods 2, 3, and 7, the resource works in the wildfire, being variable $\hat{w} = 1$. We ensure that the resource cannot fly during this time through constraint (29). During periods 4 and 6, the resource is moved to take a break due to constraint (30). Something similar happens in period 5 with constraints (27) and (28). Finally, the resource contains the wildfire in period 7 and performs the ending flight in the next period (allowed for the same reasons as the starting flight).

It is important to note that the starting period is the same for both cases since the definitions of the auxiliary variables \hat{u} are defined using s .

Table 4 represents the evolution of the wildfire and the performance of the resource over the periods. Note that the resource contains the wildfire in period 7, so from this period, the wildfire perimeter will be 0. In period 8, the resource could keep working since it would satisfy the rest policy ($\hat{w} = 1$), but the wildfire is contained, so the resource must leave it ($tr^e = 1$). To simplify the notation, in Table 4, we denote $FirePer_t$ as the perimeter of the wildfire and $ResoPer_t$ as the perimeter performed by the resources in each period, i.e., for all $t \in T$,

$$FirePer_t := \sum_{t' \in T^t} PER_{t'} \cdot y_{t-1},$$

$$ResoPer_t := \sum_{i \in \mathcal{I}, t' \in T^t} PR_{it'} \cdot w_{it'}.$$

As shown, the reformulated problem is more complicated to solve than the Original Problem since it combines integer variables and nonlinear constraints. However, the purpose is not to solve this problem but to improve its decomposability. The reformulated problem is transformed so that when Benders decomposition is applied, the nonlinear constraints (27)–(30) are linearized by fixing the variables of the Original Problem.

Table 4 Evolution of the wildfire perimeter and the contention of the resources

	1	2	3	4	5	6	7	8	9
$FirePer_t$	2.0	2.1	2.2	2.3	2.4	2.5	2.6	0.0	0.0
$ResoPer_t$	0.0	1.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0

Similar to the notation of a solution in Sect. 2.2, we define a solution for the Original Problem as:

$$SOL^* = (s_{11}^*, \dots, s_{nm}^*, tr_{11}^*, \dots, tr_{nm}^*, r_{11}^*, \dots, r_{nm}^*, er_{11}^*, \dots, er_{nm}^*, e_{11}^*, \dots, e_{nm}^*, y_1^*, \dots, y_m^*, \mu_{11}^*, \dots, \mu_{gm}^*).$$

In the case of the reformulated problem, a solution is defined similarly by adding values of the new variables at the end of the original solution vector:

$$\begin{aligned} \hat{SOL}^* &= (s_{11}^*, \dots, s_{nm}^*, tr_{11}^*, \dots, tr_{nm}^*, r_{11}^*, \dots, r_{nm}^*, er_{11}^*, \dots, er_{nm}^*, \\ &\quad e_{11}^*, \dots, e_{nm}^*, y_1^*, \dots, y_m^*, \mu_{11}^*, \dots, \mu_{gm}^*, tr_{11}^{s*}, \dots, tr_{nm}^{s*}, \\ &\quad tr_{11}^{e*}, \dots, tr_{nm}^{e*}, tr_{11}^{r*}, \dots, tr_{nm}^{r*}, \hat{tr}_{11}^{r*}, \dots, \hat{tr}_{nm}^{r*}, \\ &\quad \hat{r}_{11}^*, \dots, \hat{r}_{nm}^*, \hat{er}_{11}^*, \dots, \hat{er}_{nm}^*, \hat{e}_{11}^*, \dots, \hat{e}_{nm}^*) \\ &= (SOL^*, SOL_R^*). \end{aligned}$$

To show the equivalence of the Original Problem and the reformulated problem, the following remark is introduced.

Remark 2.1 The auxiliary variables w_{it} are nonnegative for all $i \in \mathcal{I}$ and $t \in \mathcal{T}$.

Proof This remark can be demonstrated by contradiction. First, from constraint (r18), we know that

$$\begin{aligned} \hat{r}_{it} + \hat{tr}_{it}^r &\leq \hat{u}_{it} \Rightarrow u_{it} \cdot \hat{r}_{it} + u_{it} \cdot \hat{tr}_{it}^r \leq u_{it} \cdot \hat{u}_{it} \Rightarrow \\ r_{it} + tr_{it}^r &\leq u_{it} \text{ (by definition of } w_{it}) \Rightarrow r_{it} + tr_{it}^r \leq w_{it} + r_{it} + tr_{it} \Rightarrow \end{aligned}$$

$$w_{it} \geq tr_{it}^r - tr_{it}.$$

Now, for the sake of contradiction, let us suppose that $w_{it} < 0$. Then,

$$\begin{aligned} tr_{it}^r - tr_{it} < 0 &\Rightarrow tr_{it} = 1 \text{ and } tr_{it}^r = 0 \Rightarrow \\ tr_{it}^s = 1 \text{ or } tr_{it}^e = 1 &\text{ (by equation (29))} \Rightarrow \\ w_{it} = u_{it} \cdot \hat{w}_{it} \cdot 0 = 0 &\not\leq 0. \end{aligned}$$

□

The following proposition proves the equivalence between both problems, the Original Problem and the reformulated problem.

Proposition 2.2 Let SOL^* be a feasible solution of the Original Problem; then, there exists an associated feasible solution of the reformulated problem, $\hat{SOL}^* = (SOL^*, SOL_R^*)$. Furthermore, let $\hat{SOL}^* = (SOL^*, SOL_R^*)$ be a feasible solution of the reformulated problem; then, is a feasible solution of the Original Problem.

Proof Given a feasible solution of the Original Problem, SOL^* , it is trivial to prove that it has an associated feasible solution in the reformulated problem by considering

each $i \in \mathcal{I}$, $t \in \mathcal{T}$, $\mathbf{tr}_{it}^s = \mathbf{tr}_{it}^e = \mathbf{tr}_{it}^r = \hat{\mathbf{tr}}_{it}^r = \mathbf{tr}_{it}^* = \hat{\mathbf{r}}_{it} = \mathbf{r}_{it}^*$, $\hat{\mathbf{e}}_{it} = \mathbf{e}_{it}^*$ and $\hat{\mathbf{e}}_{it} = \mathbf{e}_{it}^*$.

In addition, we will prove that a feasible solution of the reformulated problem has an associated feasible solution in the Original Problem. Let us start by proving that constraint (r4) is equivalent to constraint (4). Due to constraint (31), for each $i \in \mathcal{I}$ and for each $t \in \mathcal{T}$,

$$A_i \cdot \mathbf{w}_{it} \leq \sum_{t' \in \mathcal{T}^t} \mathbf{tr}_{it'}^s \leq \sum_{t' \in \mathcal{T}^t} \mathbf{tr}_{it'}.$$

The equivalence between constraints (r7) and (7) is also trivial.

In order to prove the equivalence between constraint (r8) and constraint (8), let us consider the case where $ITW_i = 0$ and $IOW_i = 0$ (the other case is analogous). Then, for each $i \in \mathcal{I}$ and for each $t \in \mathcal{T}$, we have

$$\mathbf{cr}_{it} := \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^t} (t - t') \cdot \mathbf{e}_{it'} - \sum_{t' \in \mathcal{T}^t} \mathbf{r}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \mathbf{er}_{it'}.$$

Now, if period $t^* \in \mathcal{T}$ where $\mathbf{e}_{it^*} = 1$ is considered, for all $t \leq t^*$,

$$\begin{aligned} \mathbf{cr}_{it} &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^t} \mathbf{r}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \mathbf{er}_{it'} \\ &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^t} \mathbf{u}_{it'} \hat{\mathbf{r}}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \mathbf{u}_{it'} \hat{\mathbf{e}}_{it'} \\ &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^t} \hat{\mathbf{r}}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \hat{\mathbf{e}}_{it'} \\ &= \hat{\mathbf{c}}\mathbf{r}_{it}, \end{aligned}$$

where the second equation is because of constraints (27) and (28). The third equality is due to the following:

- If $\hat{\mathbf{r}}_{it} = 1$, by definition of $\hat{\mathbf{w}}_{it}$ and Remark 2.1, it is clear that the variable $\hat{\mathbf{u}}_{it} = 1$. Otherwise, if $\hat{\mathbf{r}}_{it} = 0$, the equality is trivial.
- If $\hat{\mathbf{e}}_{it} = 1$, then by constraint (r10) (if $t \geq RP_i$) or by constraint (r11) (if $t < RP_i$), it can be deduced that $\hat{\mathbf{r}}_{it} = 1$. Then, applying the previous item, we know that $\hat{\mathbf{u}}_{it} = 1$. Otherwise, if $\hat{\mathbf{e}}_{it} = 0$, the equality is trivial.

In the cases where $\hat{\mathbf{u}}_{it} = 1$, considering the definitions of the auxiliary variables \mathbf{u}_{it} and $\hat{\mathbf{u}}_{it}$, we have that

$$\hat{\mathbf{u}}_{it} = \sum_{t' \in \mathcal{T}^t} s_{it'} - \sum_{t' \in \mathcal{T}^{t-1}} \hat{\mathbf{e}}_{it'} = 1 \Rightarrow \sum_{t' \in \mathcal{T}^t} s_{it'} = 1,$$

and therefore²

$$\mathbf{u}_{it} = \sum_{t' \in \mathcal{T}^t} s_{it'} - \sum_{t' \in \mathcal{T}^{t-1}} \mathbf{e}_{it'} = 1 - \sum_{t' \in \mathcal{T}^{t-1}} \mathbf{e}_{it'} = 1,$$

² Note that $\sum_{t' \in \mathcal{T}^{t-1}} \mathbf{e}_{it'} = 0$ for all $t \leq t^*$ since $\mathbf{e}_{it^*} = 1$ by assumption.

proving that if $\hat{r}_{it} = 1$ or $\hat{e}r_{it} = 1$, then $u_{it} = 1$ for all $t \leq t^*$.

Hence, we have proved that constraint (8) holds for all $t \leq t^*$,

$$0 \leq cr_{it} = \hat{c}r_{it} \leq WP_i.$$

Otherwise, if $t > t^*$, the proof is similar since the expression of $cr_{it'}$ takes the same value for periods $t' > t^*$. This is because $u_{it'} = 0$ for all $t' > t^*$,

$$\begin{aligned} cr_{it} &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^t} (t - t') \cdot e_{it'} - \sum_{t' \in \mathcal{T}^t} r_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot er_{it'} \\ &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - (t - t^*) - \sum_{t' \in \mathcal{T}^t} u_{it'} \cdot \hat{r}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot u_{it'} \hat{e}r_{it'} \\ &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - (t - t^*) - \sum_{t' \in \mathcal{T}^t} \hat{r}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \hat{e}r_{it'} \\ &= \sum_{t' \in \mathcal{T}^t} (t + 1 - t') \cdot s_{it'} - \left(\sum_{t' \in \mathcal{T}^t} s_{it'} \right) \cdot (t - t^*) - \sum_{t' \in \mathcal{T}^t} \hat{r}_{it'} - \sum_{t' \in \mathcal{T}^t} WP_i \cdot \hat{e}r_{it'} \\ &= \sum_{t' \in \mathcal{T}^{t^*}} (t^* + 1 - t') \cdot s_{it'} - \sum_{t' \in \mathcal{T}^{t^*}} \hat{r}_{it'} - \sum_{t' \in \mathcal{T}^{t^*}} WP_i \cdot \hat{e}r_{it'} \\ &= \hat{c}r_{it^*}. \end{aligned}$$

In the second equality, we use that $e_{it'} = 0$ for all $t' \neq t^*$ and constraints (27) and (28). The third equality can be proven using a procedure similar to that used for the case $t \leq t^*$. The fourth equality is because of the definition of u_{it} and the fact that $\hat{u}_{it} \geq 0$ by constraint (18), which implies that $\sum_{t' \in \mathcal{T}^t} s_{it'} = 1$. Finally, the fifth equation results from the fact that $\hat{r}_{it} = 0$ and $\hat{e}r_{it} = 0$ for all $t > t^*$.

Hence, we have proven that constraint (8) also holds for all $t > t^*$,

$$0 \leq cr_{it} = \hat{c}r_{it^*} \leq WP_i.$$

The proof of equivalence related to constraints (r9)–(r12), is similar to those already proved, but the following considerations are important:

1. The equivalence between constraint (r9) and (9) can be proven by analyzing three different situations: $t \in (-\infty, t^* - RP_i + 1]$, $t \in (t^* - RP_i + 1, t^*]$ and $t \in (t^*, \infty)$. For the proof related to constraints (r10) and (r11) one must distinguish two cases: $t \in (-\infty, t^*]$ and $t \in (t^*, \infty)$. Finally, for constraint (r12), the proof must be done differentiating between the following cases: $t \in (-\infty, t^* - TRP_i]$, $t \in (t^* - TRP_i, t^* + TRP_i]$ and $t \in (t^* + TRP_i, \infty)$.
2. Furthermore, for constraints (r9) and (r12) it is necessary to consider the optimality of the solution to demonstrate the cases in which $t \leq t^*$ and $t \leq t^* + TRP_i$, respectively.

Finally, the equivalence between constraints (r18) and (18) is trivial using Remark 2.1:

$$0 \leq w_{it} = u_{it} - r_{it} - tr_{it} \Rightarrow u_{it} \geq r_{it} + tr_{it}. \quad \square$$

Once the equivalence between the Original Problem and its reformulation has been demonstrated, we proceed to apply the BD approach to the reformulated problem.

Benders Master Problem

The master problem seeks the containment of the forest fire without acknowledging the rest periods of the resources:

$$\begin{aligned} &\text{minimize (1)} \\ &\text{subject to} \\ &\quad (2), (3), (r4), (5), (6), (r7), (13), (14), (15), (16), (17), \\ &\quad (18), (19), (31) \end{aligned}$$

$$m \cdot s_{it} + \sum_{t' \in W_{it}^-} w_{it'} \leq m, \quad \forall (i, t) \in \mathcal{S}^*, \tag{32}$$

$$m \cdot s_{it} + \sum_{t' \in TR_{it}^-} tr^r_{it'} \leq m, \quad \forall (i, t) \in \mathcal{S}^*, \tag{33}$$

$$m \cdot s_{it} + \sum_{t' \in R_{it}^-} r_{it'} \leq m, \quad \forall (i, t) \in \mathcal{S}^*. \tag{34}$$

The variables of the Benders master Problem are the original variables: s_{it} , r_{it} , er_{it} , w_{it} , tr^s_{it} , tr^r_{it} and tr^e_{it} . Moreover, \mathcal{S}^* is the set of all the tuples that represent the resources and periods, $(i, t) \in \mathcal{I} \times \mathcal{T}$, where resource i starts in period t at some iteration of the algorithm, i.e.,

$$\mathcal{S}^* := \bigcup_{v \in \mathbb{N}} \mathcal{S}(v),$$

being

$$\mathcal{S}(v) := \{(i, t) \in \mathcal{I} \times \mathcal{T} : s_{it}(v) = 1\}.$$

Using the definition of \mathcal{S}^* , cuts (32), (33) and (34) impose where a given $(i, t) \in \mathcal{S}^*$ (i is a resource and t is a starting period) cannot work, rest and travel due to rest periods, respectively. This can be accomplished using the definition of the following sets:

$$\begin{aligned} W_{it}^- &:= \{t' \in \mathcal{T} : \hat{w}_{it'}(t) = 0\}, \\ TR_{it}^- &:= \{t' \in \mathcal{T} : \hat{tr}^r_{it'}(t) = 0\}, \\ R_{it}^- &:= \{t' \in \mathcal{T} : \hat{r}_{it'}(t) = 0\}, \end{aligned}$$

where $\hat{w}_{it'}(t)$, $\hat{tr}^r_{it'}(t)$ and $\hat{r}_{it'}(t)$ are the values of working, travel due to rest, and rest variables if resource i starts in period t , respectively.

It is important to note that the needed information to build cuts (32), (33) and (34) is obtained by solving the Benders subproblem defined below.

Benders Subproblem

The subproblem seeks to establish a correct policy of breaks considering when resources begin to work. As previously stated, knowing the beginning of the resources allows computing the break periods easily. Thus, the problem is greatly simplified:

$$\begin{aligned} &\text{maximize} && \sum_{i \in \mathcal{I}, t \in \mathcal{T}} \hat{w}_{it} \\ &\text{subject to} && (r8), (r9), (r10), (r11), (r12), (r18), (26), (27), (28), (29), (30). \end{aligned}$$

In the subproblem we employ the previous definition of the variables \hat{e}_{it} , \hat{r}_{it} , \hat{tr}_{it}^r and \hat{er}_{it} , and the expressions defining \hat{u}_{it} , \hat{w}_{it} and \hat{cr}_{it} . Additionally, the variables associated with the master problem (complicating variables) are fixed: $s_{it} = s_{it}^*$, $r_{it} = r_{it}^*$, $er_{it} = er_{it}^*$, $w_{it} = w_{it}^*$, $tr^s_{it} = tr^{s*}_{it}$, $tr^r_{it} = tr^{r*}_{it}$ and $tr^e_{it} = tr^{e*}_{it}$. Therefore, as these variables are fixed, constraints (27)–(30) become linear.

Remark 2.3 Since the subproblem only accounts for the feasibility of the solution given by the master problem regarding rest policies, we assume that every resource will work in the wildfire containment the maximum periods allowed. This is necessary for the proper performance of the resources in the subproblem.

Remark 2.4 As the structure of the subproblem shows, it is easy to check that the subproblem can be decomposed trivially by resources. We can do this because the constraints and objective functions do not share information on more than one resource. Thus, instead of solving the Benders subproblem, we can solve for each selected resource i (given from a solution of the Benders master problem). In addition, in order to obtain information on the resolution of these subproblems, we can remove constraints (27)–(30) to obtain information about the configuration of the rest periods. By doing so, the subproblem associated with resource i , which we will name **Benders subproblem i** , determines the maximum performance of resource i considering when it starts and the restrictions on rest period legislation.

Benders Algorithm

The pseudocode presented in Algorithm 1 defines the steps in the Benders algorithm in the context of our decomposition of the problem.

Algorithm 1 Adaptation of the Benders Algorithm

```

1:  $v \leftarrow 0$ 
2: while no new cuts do
3:    $v \leftarrow v + 1$ 
4:   Solve Benders Master Problem
5:   Get  $\mathcal{S}(v)$ 
6:   for  $(i, t)$  in  $\mathcal{S}(v)$  do
7:     if  $(i, t)$  not in  $\mathcal{S}^*$  then
8:       Solve Benders Subproblem  $i$ 
9:       Update  $\mathcal{S}^*$ ,  $W_{it}^-$ ,  $TR_{it}^-$  and  $R_{it}^-$ 
10:      Add the associated cuts given by constraints (32), (33) and (34)

```

It is important to note that three new types of cuts, (32), (33) and (34), are created to improve the convergence of the algorithm. The following remark will discuss the validity of the proposed cuts.

Remark 2.5 Constraints (32)–(34) are valid Benders cuts that allow us to give to the Benders master problem information of how a resource must act knowing its start period.

For a given $i \in \mathcal{I}$ and $t^s \in \mathcal{T}$ where $s_{it^s} = 1$, Benders subproblem i gives the configuration of work, travel associated with rest, and rest periods of resource i from its start period (t^s) until the last period (m). The configuration of work, travel and rest periods guarantees the feasibility of the Benders subproblem i (and consequently, the feasibility of the Benders subproblem).

Examining Benders master problem, constraint (32) fixes the working periods to 0 according to the solution of Benders subproblem i with $s_{it^s} = 1$. In a similar way, we can impose periods of no travel due to rest and periods of no rest using constraints (33) and (34), respectively.

If a solution proposed by the Benders master problem is infeasible for the reformulated problem, then the cuts generated in the next iteration will remove the solution as each Benders subproblem i will compute a feasible configuration for the resource i to construct the new cuts. Otherwise, if the solution is feasible for the reformulated problem, then it will be feasible for each Benders subproblem i and the associated cuts will admit it.

3 Fixed activity benders problem

Inspired by the idea of BD about pre-computing the periods in which resources must rest, we propose a new formulation of the problem that guarantees convergence to the global optimum.

Suppose the period in which a resource begins to work in the wildfire and their initial conditions are known. In that case, it is possible to calculate the work and rest periods

according to the pertinent legislation. Therefore, we can simplify the BD method, computing in advance this information without solving the Benders subproblems.

As described above in the proposed BD method, the Benders subproblems share the primal solutions obtained with the Benders master problem with the aim of defining iteratively new cuts (32)–(34) removing solutions that do not verify the policy of breaks.

In this way, we propose a new problem formulation equivalent to the Benders master problem where all the feasibility cuts are defined a priori without the need of solving any subproblem. In order to pre-compute these feasibility cuts efficiently, a tailor-made algorithm (see Algorithm 2) has been designed that takes advantage of the specific features of the problem. Note that this new formulation requires to introduce all the feasibility cuts at once, so the size of the resulting problem is bigger than the ones arising in the decomposition procedures. Although this could derive in a significant computationally burden when solving the problem, the computational results that will be presented later shows a good behaviour of this method in practice, outperforming the other decomposition approaches.

The new problem formulation can be stated as follows:

Fixed Activity Benders Problem

$$\begin{aligned} &\text{minimize (1)} \\ &\text{subject to} \\ &\quad (2), (3), (r4), (5), (6), (r7), (13), (14), (15), (16), (17), \\ &\quad (18), (19), (31) \end{aligned}$$

$$m \cdot s_{it} + \sum_{t' \in W_{it}^-} w_{it'} \leq m, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (r32)$$

$$m \cdot s_{it} + \sum_{t' \in TR_{it}^-} tr_{it'} \leq m, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, \quad (r33)$$

$$m \cdot s_{it} + \sum_{t' \in R_{it}^-} r_{it'} \leq m, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}. \quad (r34)$$

Therefore, before creating the Fixed Activity Benders Problem, we need to compute sets W_{it}^- , TR_{it}^- and R_{it}^- (see Algorithm 2 for more details) to create constraints (r32)–(r34), respectively.

Algorithm 2 initializes a counter for the work periods (*WorkCounter*) for a given resource and start period. Suppose the resource was previously working in any wildfire ($ITW_i = 1$ or $IOW_i = 1$). This implies that two situations might occur: if the resource starts to work in the first period, the counter is updated with its initial information; and if the resource is resting, it cannot be selected during the wildfire’s extinguishing.

Once all variables have been initialized, the counter is iteratively increased. Thus, when it reaches $WP_i - TRP_i$ working periods, the resource will perform the needed

Algorithm 2 Compute work, travel and rest periods of a resource knowing its start period

Input: $i^* \in \mathcal{I}$ (selected resource), $t^* \in \mathcal{T}$ (start period)

```

1:  $WorkCounter \leftarrow 1$ 
2: if  $ITW_i = 1$  or  $IOW_i = 1$  then
3:   if  $t^* = 1$  then
4:      $WorkCounter \leftarrow CWP_i - CRP_i + 1$ 
5:   else if  $RP_i \neq 0$  then
6:      $WorkCounter \leftarrow WP_i + 1$ 
7:      $t^* \leftarrow m + 1$  (avoid starting working,  $t^* \geq m$ )
8:  $Work_{it}, Travel_{it}, Rest_{it} \leftarrow 0$ , for all  $i \in \mathcal{I}, t \in \mathcal{T}$ 
9:  $TravelCounter \leftarrow 0$ 
10: for  $t$  in  $\mathcal{T}_{t^*}$  do
11:   if  $t \geq m - TRP_{i^*} + 1$  then
12:      $WorkCounter \leftarrow WorkCounter + 1$ 
13:      $TravelCounter \leftarrow TravelCounter + 1$ 
14:      $Travel_{it} \leftarrow 1$ 
15:     continue
16:   else if  $WorkCounter \leq WP_i - TRP_i$  then
17:      $WorkCounter \leftarrow WorkCounter + 1$ 
18:      $Work_{it} \leftarrow 1$ 
19:   else if  $WorkCounter \leq WP_i$  then
20:      $WorkCounter \leftarrow WorkCounter + 1$ 
21:      $TravelCounter \leftarrow TravelCounter + 1$ 
22:      $Travel_{it} \leftarrow 1$ 
23:   else if  $WorkCounter \leq WP_i + RP_i$  then
24:      $WorkCounter \leftarrow WorkCounter + 1$ 
25:      $Rest_{it} \leftarrow 1$ 
26:   else if  $WorkCounter \leq WP_i + RP_i + TRP_i$  then
27:      $WorkCounter \leftarrow WorkCounter + 1$ 
28:      $TravelCounter \leftarrow TravelCounter + 1$ 
29:      $Travel_{it} \leftarrow 1$ 
30:   else
31:      $WorkCounter \leftarrow 1 + TRP_i$ 
32:      $TravelCounter \leftarrow TravelCounter + 1$ 
33:      $Travel_{it} \leftarrow 1$ 
return  $Work, Travel, Rest$ 

```

periods to rest. In this case, when the resting phase is finished, the counter is reset, and the procedure described above is repeated until the last period.

In addition, it is important to note that this new method allows incorporating on-demand rest policies, facilitating their adaptation to more complex contexts.

4 Computational results

This section studies the impact of the decomposition techniques explained above, using a representative set of simulated instances of the original wildfire extinction model. To simplify the understanding of this work, we use the following nomenclature:

- **OP**: Original Problem (Sect. 1.2).
- **AL**: A strategy based on Augmented Lagrangian decomposition (Sect. 2.1).
- **BP**: Branch and Price decomposition provided by SCIP (Sect. 2.2).

- **BD**: Benders Decomposition (Sect. 2.3).
- **FA**: Fixed Activity Benders problem (Sect. 3).

The experiments were conducted using the Finisterrae II supercomputer, provided by Galicia Supercomputing Centre (CESGA), which consists of 306 nodes powered with two deca-core Intel Haswell 2680v3 CPUs with 128 GB of RAM connected through an Infiniband FDR network. The implementations were programmed in Python version 3.6.8. All the MILP models were solved using Gurobi (version 8.1.0, Gurobi Optimization (2020)), except in the case of BP, where we employed SCIP (version 6.0.1, Gleixner et al. (2018)) framework.

In order to compare the algorithms, we generated two sets of instances (see “Appendix B” for details about the instance generator). The first, used in Sect. 4.1, corresponds to small instances where all the algorithms can provide solutions for most cases. The second, used in Sect. 4.2, contains larger instances based on realistic case studies. More details about the sizes, composition and periods in instances are in the tables of “Appendix C”.

Regarding the experimental settings, all optimization runs used a stopping criterion based on a predefined computational time of 10 min. The goal is to compare the solutions obtained, in terms of the objective function, using the different techniques at the same threshold time. We selected this time considering the requirements of wildfire contention services in real situations. About configuration settings of the methods, just to comment that AL sets the parameters β and λ to 0.3 and 1000, respectively.

Due to the high number of experiments conducted, we have decided to summarize the results over classes of instances with the same problem size. In detail, we present boxplot graphs for the relative objective functions and the relative computational times. Namely, for each instance, let x' be the best solution found, the relative objective function associated to a solution x is calculated as the quotient $f(x)/f(x')$ (analogously for the computational time). Thus, values closer to 1 represent better performance.

In addition, we show different tables with a summary of statistical results (minimum, mean, median, maximum, standard deviation and percentage of solved instances). For readers more familiar with performance profile graphs (Dolan and Moré 2002), we also provide them for the objective function and for the computational time in “Appendix D”.

The code and instances required to reproduce the results are available at:

<https://github.com/jorgerodriguezveiga/firedecomp>

4.1 Performance analysis in small instances

In order to analyze the proposed solution techniques, we defined 16 different groups of small instances by combining resource groups (brigades, aircraft and machines) in different ways: using 2 or 4 members per group and setting the number of time periods to 10 or 15. Furthermore, for each group, we randomly generated 100 instances, which gave us a total of 1600 instances. The details of the size (numbers of variables and constraints) of the different groups of instances are shown in Table 9.

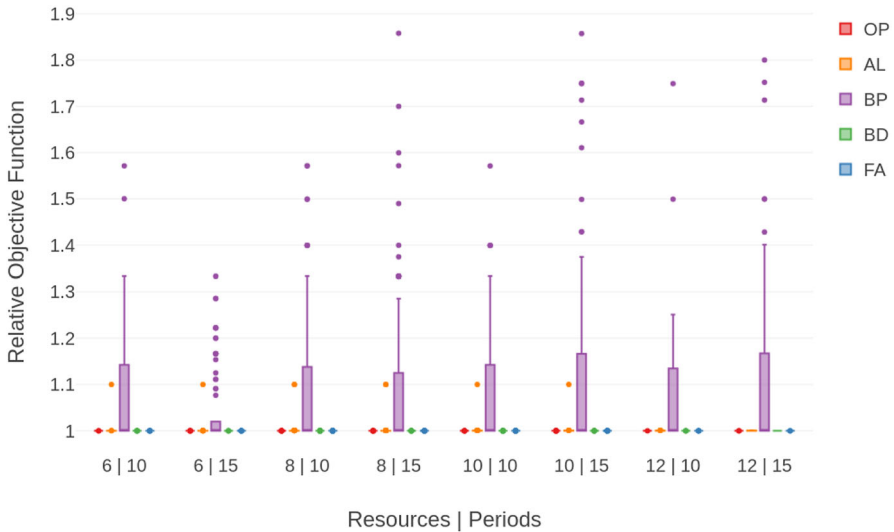


Fig. 2 Boxplot of the relative objective function for small instances

Table 5 Relative objective function statistics for small instances

	min	mean	median	max	sd	solved (%)
FA	1.0000	1.0000	1.0000	1.0000	0.0000	100.0000
BD	1.0000	1.0000	1.0000	1.0000	0.0000	100.0000
AL	1.0000	1.0007	1.0001	1.1001	0.0071	100.0000
BP	1.0000	1.0709	1.0002	1.8578	0.1257	91.6933
OP	1.0000	1.0000	1.0000	1.0000	0.0000	100.0000

As stated before, all the instances were solved with our techniques in a maximum time of 10 min. Thus, for each run, we have collected the objective function obtained during the optimization and the time required to reach this solution within the time threshold.

Figure 2 is a boxplot where we compare our methods in different classes that include various groups of instances. In this way, it shows on its x-axis the number of resources and periods of the problem, separated by a vertical bar (|), and on its y-axis, the relative objective function. The results reveal that BP exhibits the worst performance in terms of the objective function.³ The other methodologies have a similar behaviour always reaching the best optimal solution, except AL, which fails to do it in some specific instances. These conclusions are also corroborated in Table 5, where interestingly, it can be seen that all techniques are able to provide feasible solutions for all the instances, except BP.

³ The performance profile for the objective function can be seen in 6.

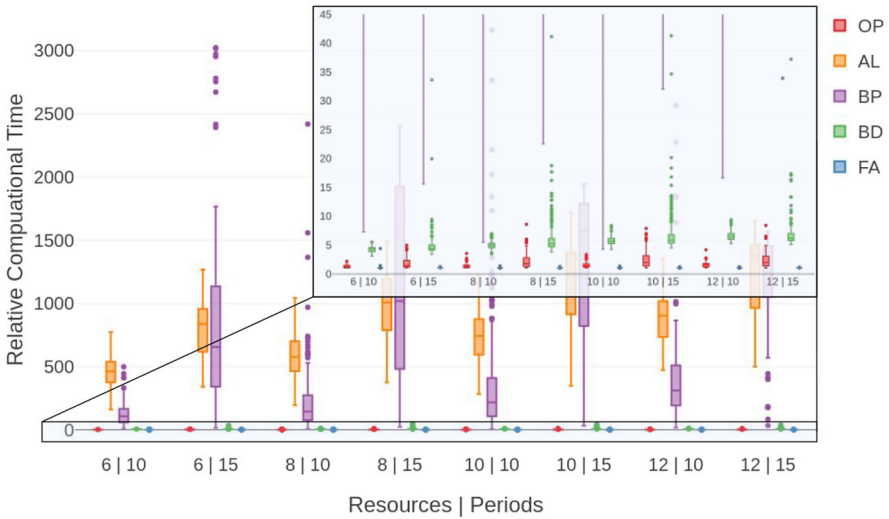


Fig. 3 Boxplot of the relative computational time for small instances

Table 6 Relative computational time statistics for small instances

	min	mean	median	max	sd	solved (%)
FA	1.0000	1.0126	1.0000	4.3725	0.0946	100.0000
BD	3.0529	5.8480	5.3920	46.7558	2.7328	100.0000
AL	161.1682	862.1996	811.2871	1719.7460	329.8117	100.0000
BP	4.2376	732.4274	431.5065	3164.3300	695.1925	91.6933
OP	1.0000	1.7972	1.4329	8.5698	0.9883	100.0000

Figure 3 and Table 6 represent the results concerning the relative computational time.⁴ We can see that FA, BD and OP, are the fastest ones with a big difference with respect to AL and BP. Further, the zoom over the figure (and measures in Table 6) shows how FA outperforms BD and OP. Additionally, the boxplot of the absolute computational times for each group of instances can be seen in Fig. 10.

In conclusion, for small instances, the algorithm that performs the best is FA. Although OP and BD exhibit a good behaviour in terms of optimal values, their computational times are slightly higher. Further, the AL and BP algorithms have much longer computational times, and BP does not converge in many cases.

⁴ The performance profile for the computational time can be seen in 7.

4.2 Performance analysis in realistic-scale instances

In this case, we have defined 48 different groups of realistic-scale instances (see Table 10 for details), generating 100 random instances for each group, which gives us a total of 4800 instances.

Based on the results for the small instances, where OP and FA exhibit the best performance, we only analyze the behaviour of these two methods over these larger instances. In fact, as the time threshold is set again to 10 min, we have checked that the other algorithms were not able to reach a feasible solution for most of these instances.

Figure 4 is analogous to the previous ones. In this case, we have split the instances into 24 classes according to the number of resources and periods. The results obtained in the relative objective function for both methods demonstrate that FA finds for all instances the best optimum while OP only reaches it for the smaller ones, those with less than 60 periods and 30 resources.⁵ The same conclusions can be clearly seen looking at the summaries presented in Table 7.

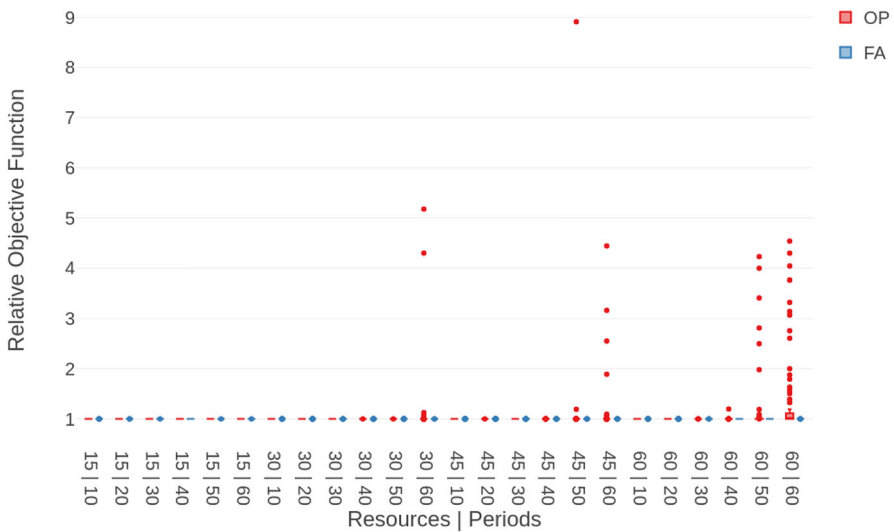


Fig. 4 Boxplot of the relative objective function for realistic-scale instances

Table 7 Relative objective function statistics for realistic-scale instances

	Min	Mean	Median	Max	SD	Solved (%)
FA	1.0000	1.0000	1.0000	1.0000	0.0000	100.0000
OP	1.0000	1.0142	1.0000	8.9111	0.2098	97.3739

⁵ The performance profile for the objective function can be seen in Fig. 8.

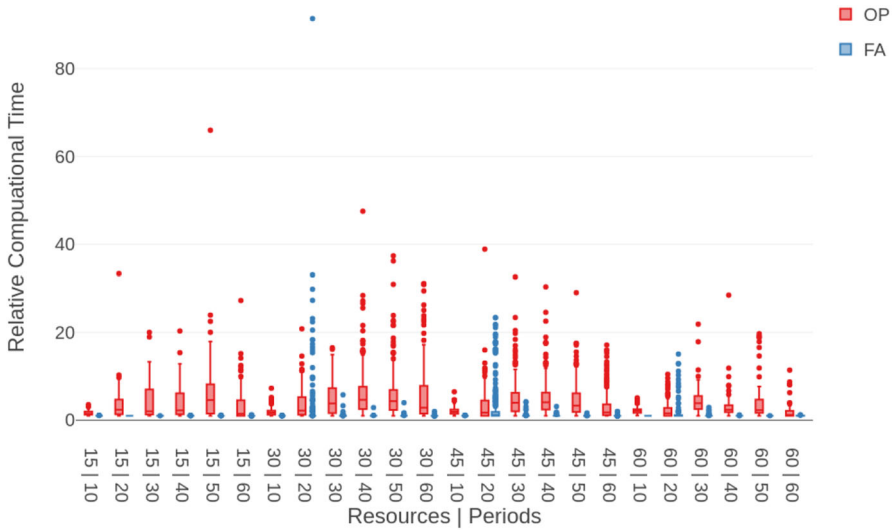


Fig. 5 Boxplot of the relative computational time for realistic-scale instances

Table 8 Relative computational time statistics for realistic-scale instances

	Min	Mean	Median	Max	SD	Solved (%)
FA	1.0000	1.2523	1.0000	91.3941	2.1800	100.0000
OP	1.0000	4.0239	2.4890	65.9690	4.0681	97.3739

Regarding the relative computational time of the algorithms, Fig. 5 shows that FA exhibits again the best performance, being faster than OP.⁶ More in detail, in Table 8, we can see that FA is at least 2.5 faster than OP in 50% of the instances (the median of the OP is 2.4890). Additionally, the boxplot of the absolute computational times for each group of instances can be seen in Fig. 11.

5 Conclusions

In this study, we have proposed different decomposition techniques to efficiently tackle a wildfire suppression model to manage the operations of resources.

Implementing decomposition techniques is not usually a straightforward task since it requires adapting the technique to the specific features of the problem. Thus, we have proposed a modification of the augmented Lagrangian decomposition to achieve convergence of our specific problem. Moreover, we have applied Benders and branch and price decompositions to our problem. In the case of Benders, to improve convergence to an optimal point, we have modified the decomposition and proposed new cuts specific to this problem.

⁶ The performance profile for the computational time can be seen in Fig. 9.

Furthermore, the analysis of decomposition methods has encouraged us to propose the so-called Fixed Activity Benders Problem. It is a reformulation of the original model, where some information about resources (such as rest, travel and working time) can be precomputed.

An extensive computational study comparing the different proposed alternatives using a set of randomly generated instances of the wildfire extinguishment model has been conducted. In the results, we observe that the Fixed Activity Benders Problem clearly outperforms the rest of the techniques in terms of optimality and computational times.

As future work, it could be interesting to seek alternative decomposition of the problem that allows us to apply the same or different decomposition techniques more efficiently. Furthermore, it might be promising to study how the Fixed Activity Benders Problem can be embedded in such a decomposition framework.

Acknowledgements This research work is supported by the R+D+I project grants PID2020-116587GB-I00 and PID2021-124030NB (C31 and C32), funded by MCIN/AEI/10.13039/501100011033/ and by “ERDF A way of making Europe”/EU. Second author investigation is funded by the Xunta de Galicia (contract post-doctoral 2019-2022). We acknowledge the computational resources provided by CESGA. Third author acknowledges support from the Xunta de Galicia through the ERDF (ED431C-2020-14 and ED431G 2019/01), and “CITIC”.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A. Augmented Lagrangian implementations details

Given a resource i , the associated **Augmented Lagrangian Subproblem i** is formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{t \in \mathcal{T}} C_i \cdot \mathbf{u}_{it} + P_i \cdot \mathbf{z}_i + \sum_{t \in \mathcal{T}} NVC_t \cdot \bar{y}_{t-1} + \sum_{g \in \mathcal{G}, t \in \mathcal{T}} M' \cdot \boldsymbol{\mu}_{gt} \\ & + (\bar{\lambda}_1 \cdot (\text{penSet}_1 + v_1) + \frac{1}{2} \cdot \bar{\beta}_1 \cdot (\text{penSet}_1 + v_1)^2) \\ & + \sum_{t \in \mathcal{T}} (\bar{\lambda}_{t+1} \cdot (\text{penSet}_2(t) + v_{t+1}) + \frac{1}{2} \cdot \bar{\beta}_{t+1} \cdot (\text{penSet}_2(t) + v_{t+1})^2) \end{aligned}$$

subject to

$$\begin{aligned} & (4)–(19) \\ & v_j \geq 0, \quad \forall j \in \{1, \dots, t + 1\}, \end{aligned}$$

being

$$\begin{aligned}
 penSet_1 &= \sum_{t \in \mathcal{T}} PER_t \cdot \bar{y}_{t-1} - \sum_{t \in \mathcal{T}} (PR_{it} \cdot w_{it} + \sum_{i' \in \mathcal{I}, i' \neq i} PR_{i't} \cdot \bar{w}_{i't}), \\
 penSet_2(t) &= -M \cdot \bar{y}_t + \sum_{i' \in \mathcal{I}^t} (PER_{i'} \cdot \bar{y}_{t-1}) \\
 &\quad - \sum_{i' \in \mathcal{I}^t} (PR_{i't'} \cdot w_{i't'} + \sum_{i'' \in \mathcal{I}, i'' \neq i'} PR_{i''t'} \cdot \bar{w}_{i''t'}),
 \end{aligned}$$

where $\bar{\lambda}$ corresponds to fixed Lagrange multipliers, $\bar{\beta}$ is the fixed penalty parameters large enough to ensure local convexity, and \bar{y} is the contained period of the wildfire. Since each subproblem needs to have coherence in variable y to guarantee the separability, we have decided to fix the variable as a constant, and explore all the possibilities of y for each iteration of the ALD method. Moreover, expressions $penSet_1$ and $penSet_2$ represent inequality constraints (2) and (3), respectively, which are modified to appear penalized in the objective function. v is a set of auxiliary variables to transform inequalities $penSet_1$ and $penSet_2$ into equality constraints. As can be observed, the ALD method adds quadratic terms to penalties to confer good convexity properties.

Regarding the formulation of $penSet_1$ and $penSet_2$, we highlight that w_{it} are primary variables of subproblem i that represent the working time of resource i , and $\bar{w}_{i't}$ are constant values that correspond to the values of w in other resources. Thus, the value of each resource i' will be addressed in its respective subproblem, consequently facilitating the decomposition of the Original Problem.

Theoretically, although constraints (14) and (15) could also be penalized in the objective function, we have empirically observed that it causes the convergence to worsen, making it more complex to coordinate their penalties in the resulting subproblems.

As stated above, the ALD method is an iterative algorithm, that seeks to calibrate their penalties to obtain reasonable feasible solutions. It requires an initial solution that can be obtained, for example, using a constructive heuristic. Then, the algorithm repeats the following steps until a stopping criterion is fulfilled. The algorithm decomposes the Original Problem, iterates over each period $t \in \mathcal{T}$, fixes $y_t = 0$ to consider it as the possible contained period, and solve the corresponding subproblem with a specific λ and β . These parameters are updated with the results provided by the subproblem in each iteration. This calibration process is conducted using a subgradient method, which is a strategy widely used in the AL literature. Although using the subgradient method causes the loss of the global optimality guarantee in MILP problems (Fisher 2004), the algorithm exhibited good performance in our experiments. The following expressions represent an example of how to update these multipliers and constants in the iteration k and for penalty $PenSet_1$:

$$\begin{aligned}
 \lambda_1^{(k+1)} &= \lambda_1^{(k)} + \beta_1^{(k)} \cdot PenSet_1^{(k)}, \\
 \beta_1^{(k+1)} &= 1.2 \cdot \beta_1^{(k)}.
 \end{aligned}$$

$PenSet_2$ also has an analogous procedure. Moreover, the method updates the fixed resources' value in the subproblems at the end of each iteration, but only when a new feasible solution is reached. Finally, ALD will stop when it has converged or has obtained an infeasible problem for each possible y .

B. Description of instances generator

We have implemented an instance generator intending to create a test set to analyze the decomposition methods described throughout the work. The following paragraphs describe the values and intervals of parameters to simulate instances in the wildfire extinction model.

First, the wildfire extinguishing period considered in all instances has been 10 min, classified the resources into three groups: brigades, aircraft and machines. The minimum resource number per group is between 1 and a quarter of that type's number of resources. And the maximum is between the number of resources of that type and three-quarters of it.

Moreover, each sort of resource has unique characteristics, such as in the case of performance, where brigades have values between 2–10 km/h, while aircraft and machines are between 4–15 km/h. In terms of costs, all have a maximum fixed cost of 1000 €. The variable costs depend on the type of resource: 200–500 €/h for brigades; 1000–3000 €/h for aircraft; and 500–1000 €/h in the case of machines.

Likewise, parameters according to the maximum work and rest periods depend on legislation. In these cases, we have taken into consideration Spanish law regulations. It indicates that all resources have a maximum daily time of 480 min (8 h) and a time between breaks of 10 min. Furthermore, there is a maximum working time without breaks of 120 min in the aircraft case, with a rest time of 40 min.

All resources have a low probability of operating already in the wildfire, a medium probability of working in another wildfire, and a high probability of being idle. If they are not in the wildfire, they must travel there, being the arrival times 30–120 min for brigades, 10–40 min for aircraft, and 60–180 min in the case of machines. Additionally, if a resource had been active in the same wildfire, its previous work time must be considered. It implies that the generator will select a value between 0 and 470 min.

On the other hand, the wildfire increases its perimeter between 1–3 km every 10 min. As the cost depends on the fire's spread, it is calculated by multiplying a factor by the perimeter increment. This factor has been chosen within the interval of 300–500. Further, to ensure the feasibility, the generator has considered the resource's performance to calculate the perimeter.

Last, the instance generator has always considered the efficiency of the resources as one in all periods.

The characterization above describes the general behaviour of the instance generator function. To see details, we recommend examining the GitHub repository.

C. Dimensions of the instances

See Tables 9, 10.

Table 9 Dimensions of the problems

Brigades	Aircraft	Machines	Periods	Variables	Constraints	Class
2	2	2	10	340	521	6 10
2	2	2	15	510	766	6 15
2	2	4	10	440	671	8 10
2	2	4	15	660	986	8 15
2	4	2	10	440	671	8 10
2	4	2	15	660	986	8 15
2	4	4	10	540	821	10 10
2	4	4	15	810	1206	10 15
4	2	2	10	440	671	8 10
4	2	2	15	660	986	8 15
4	2	4	10	540	821	10 10
4	2	4	15	810	1206	10 15
4	4	2	10	540	821	10 10
4	4	2	15	810	1206	10 15
4	4	4	10	640	971	12 10
4	4	4	15	960	1426	12 15

Table 10 Dimensions of the problems in realistic-scale cases

Brigades	Aircraft	Machines	Periods	Variables	Constraints	Class
5	5	5	10	790	1196	15 10
5	5	5	20	1580	2316	15 20
5	5	5	30	2370	3436	15 30
5	5	5	40	3160	4556	15 40
5	5	5	50	3950	5676	15 50
5	5	5	60	4740	6796	15 60
5	5	20	10	1540	2321	30 10
5	5	20	20	3080	4491	30 20
5	5	20	30	4620	6661	30 30
5	5	20	40	6160	8831	30 40
5	5	20	50	7700	11,001	30 50
5	5	20	60	9240	13,171	30 60
5	20	5	10	1540	2321	30 10
5	20	5	20	3080	4491	30 20
5	20	5	30	4620	6661	30 30

Table 10 continued

Brigades	Aircraft	Machines	Periods	Variables	Constraints	Class
5	20	5	40	6160	8831	30 40
5	20	5	50	7700	11, 001	30 50
5	20	5	60	9240	13, 171	30 60
5	20	20	10	2290	3446	45 10
5	20	20	20	4580	6666	45 20
5	20	20	30	6870	9886	45 30
5	20	20	40	9160	13, 106	45 40
5	20	20	50	11, 450	16, 326	45 50
5	20	20	60	13, 740	19, 546	45 60
20	5	5	10	1540	2321	30 10
20	5	5	20	3080	4491	30 20
20	5	5	30	4620	6661	30 30
20	5	5	40	6160	8831	30 40
20	5	5	50	7700	11, 001	30 50
20	5	5	60	9240	13, 171	30 60
20	5	20	10	2290	3446	45 10
20	5	20	20	4580	6666	45 20
20	5	20	30	6870	9886	45 30
20	5	20	40	9160	13, 106	45 40
20	5	20	50	11, 450	16, 326	45 50
20	5	20	60	13, 740	19, 546	45 60
20	20	5	10	2290	3446	45 10
20	20	5	20	4580	6666	45 20
20	20	5	30	6870	9886	45 30
20	20	5	40	9160	13, 106	45 40
20	20	5	50	11, 450	16, 326	45 50
20	20	5	60	13, 740	19, 546	45 60
20	20	20	10	3040	4571	60 10
20	20	20	20	6080	8841	60 20
20	20	20	30	9120	13, 111	60 30
20	20	20	40	12, 160	17, 381	60 40
20	20	20	50	15, 200	21, 651	60 50
20	20	20	60	18, 240	25, 921	60 60

D. Auxiliary figures

See Figs. 6, 7, 8, 9, 10 and 11.

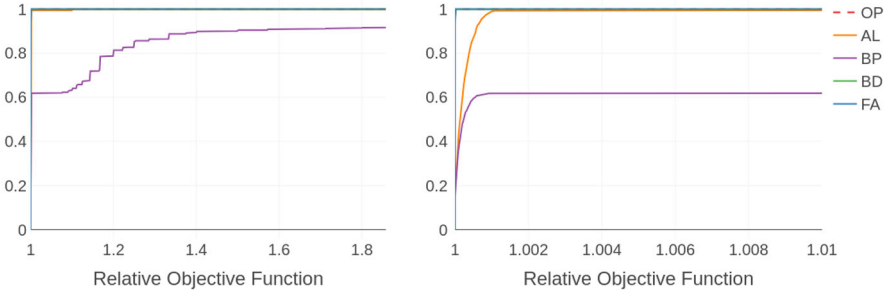


Fig. 6 Performance profile of the objective function on the intervals [1, 1.8] (left) and [1, 1.01] (right) for small instances

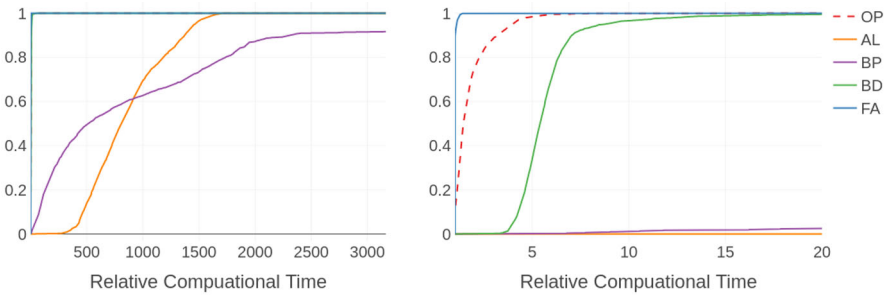


Fig. 7 Performance profile of the computational time on the intervals [1, 3000] (left) and [1, 20] (right) for small instances

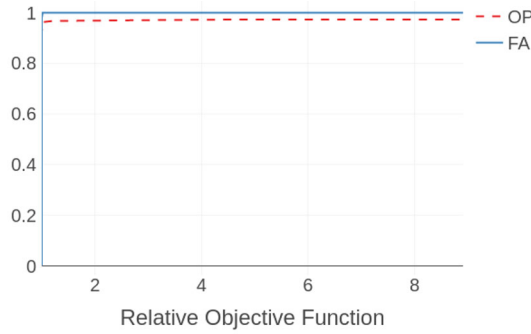


Fig. 8 Performance profile of the objective function on the interval [1, 8] for realistic-scale instances

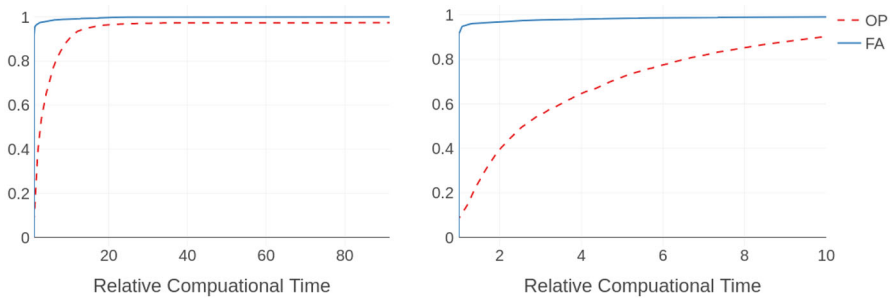


Fig. 9 Performance profile of the computational time on the intervals [1, 90] (left) and [1, 10] (right) for realistically sized instances

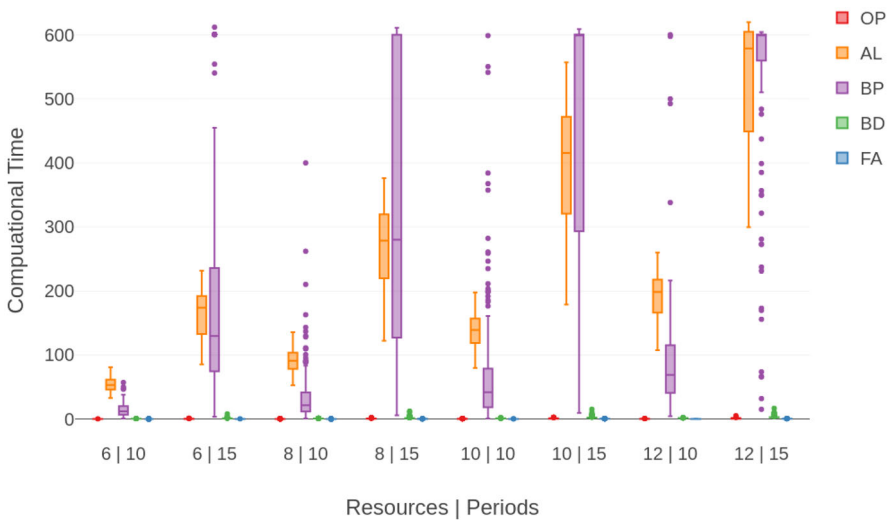


Fig. 10 Boxplot of the computational time of the algorithm only considering the solved instances for small cases. X axis represent the size of the instances, indicating the number of resources and periods separated by “|”, respectively

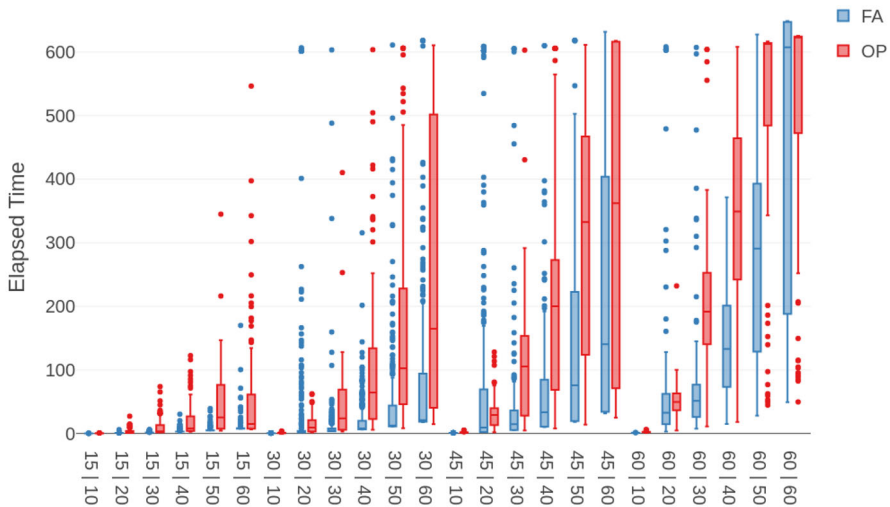


Fig. 11 Boxplot of the computational time of the algorithm only considering the solved instances for realistic-scale cases. X axis represent the size of the instances, indicating the number of resources and periods separated by “|”, respectively

References

- Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: column generation for solving huge integer programs. *Oper Res* 46:316–329
- Benders J (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer Math* 4:238–252
- Caunhye AM, Nie X, Pokharel S (2012) Optimization models in emergency logistics: a literature review. *Socioecon Plann Sci* 46:4–13
- Commission European (2020) Forest fires in Europe, Middle East and North Africa 2019. Technical report, JRC technical reports
- Conejo A, Castillo E, Mínguez R, García-Bertrand R (2006) Decomposition techniques in mathematical programming: engineering and science applications. Springer
- Cordeau JF, Stojković G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transp Sci INFORMS* 35:375–388
- Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. *Math Program* 91:201–213
- Donovan GH, Rideout DB (2003) An integer programming model to optimize resource allocation for wildfire containment. *For Sci* 49:331–335
- Finney MA (1998) FARSITE: fire area simulator—model development and evaluation, vol 3. United States Department of Agriculture, Forest Service, Rocky Mountain Research Station
- Fisher ML (2004) The Lagrangian relaxation method for solving integer programming problems. *Manag Sci* 50:1861–1871
- Gleixner A, Bastubbe M, Eifler L, Gally T, Gamrath G, Gottwald RL, Hendel G, Hojny C, Koch T, Lübbecke ME, Maher SJ, Miltenberger M, Müller B, Pfetsch ME, Puchert C, Rehfeldt D, Schlösser F, Schubert C, Serrano F, Shinano Y, Viernickel JM, Walter M, Wegscheider F, Witt JT, Witzig J (2018) The SCIP optimization suite 6.0. Technical report. Optimization online. http://www.optimization-online.org/DB_HTML/2018/07/6692.html
- Gorte JK, Gorte RW (1979) Application of economic techniques to fire management—a status review and evaluation. Gen. Tech. Rep. INT-GTR-53. US Department of Agriculture, Forest Service, Intermountain Research Station, Ogden, vol 26, p 53
- Gurobi Optimization L (2020) Gurobi optimizer reference manual. <http://www.gurobi.com>

- Hamdi A, Mishra SK (2011) Decomposition methods based on augmented Lagrangians: a survey. In: Topics in nonconvex optimization. Springer, pp 175–203
- Headley R (1916) Fire suppression, district 5. USDA-Forest Service, pp 1–57
- Martell DL (2015) A review of recent forest and wildland fire management decision support systems research. *Curr For Rep* 1:128–137
- Miller C, Ager AA (2013) A review of recent advances in risk analysis for wildfire management. *Int J Wildland Fire* 22:1–14
- Minas JP, Hearne JW, Handmer JW (2012) A review of operations research methods applicable to wildfire management. *Int J Wildland Fire* 21:189–196
- Ntaimo L, Arrubla JAG, Stripling C, Young J, Spencer T (2012) A stochastic programming standard response model for wildfire initial attack planning. *Can J For Res* 42:987–1001
- Papadakos N (2009) Integrated airline scheduling. *Comput Oper Res* 36:176–195
- Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The benders decomposition algorithm: a literature review. *Eur J Oper Res* 259:801–817. <https://doi.org/10.1016/j.ejor.2016.12.005>
- Rios J, Ross K (2010) Massively parallel Dantzig–Wolfe decomposition applied to traffic flow scheduling. *J Aerosp Comput Inf Commun* 7:32–45
- Rodríguez-Veiga J, Ginzo-Villamayor MJ, Casas-Méndez B (2018) An integer linear programming model to select and temporally allocate resources for fighting forest fires. *Forests* 9:583
- Romanski J, Hentenryck PV (2016) Benders decomposition for large-scale prescriptive evacuations. In: AAAAI'16: Proceedings of the thirtieth AAAI conference on artificial intelligence, pp 3894–3900
- Sethi S, Sorger G (1991) A theory of rolling horizon decision making. *Ann Oper Res* 29:387–415
- Sparhawk WN (1925) The use of liability ratings in planning forest fire protection. National Emergency Training Center, Emmitsburg
- Vanderbeck F, Savelsbergh MW (2006) A generic view of Dantzig–Wolfe decomposition in mixed integer programming. *Oper Res Lett* 34:296–306
- Vanderbeck F, Wolsey LA (1996) An exact algorithm for IP column generation. *Oper Res Lett* 19:151–159
- Wei Y, Rideout DB, Hall TB (2011) Toward efficient management of large fires: a mixed integer programming model and two iterative approaches. *For Sci* 57:435–447
- Xunta de Galicia (2017) PLADIGA: Memoria. http://mediorural.xunta.gal/fileadmin/arquivos/forestal/pladiga/2017/2_MEMORIA.pdf. Accessed 19 Dec 2022 17:41:37

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.