European Journal of Operational Research 304 (2023) 887–900

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Iterated local search for the placement of wildland fire suppression resources

André Bergsten Mendes[a,*], Filipe Pereira e Alvelos[b]

[a] *Departamento de Engenharia Naval e Oceânica, Universidade de São Paulo Av. Prof. Mello Moraes 2231, 05508-030, São Paulo, Brasil*
[b] *Centro Algoritmi/Departamento de Produção e Sistemas, Universidade do Minho, 4710-057, Braga, Portugal*

ARTICLE INFO

Article history:
Received 26 May 2021
Accepted 22 April 2022
Available online 28 April 2022

Keywords:
Metaheuristics
Wildfires
Fire suppression
Mixed integer programming
Iterated local search

ABSTRACT

We consider the problem of, given a landscape represented by a gridded network and a fire ignition location, deciding where to locate the available fire suppression resources to minimise the burned area and the number of deployed resources as a secondary objective. We assume an estimate of the fire propagation times between adjacent nodes and use the minimum travel time principle to model the fire propagation at a landscape-level. The effect of locating a resource in a node is that it becomes protected and the fire propagation to its unburned adjacent nodes is delayed. Therefore, the problem is to identify the most promising nodes to locate the resources, which is solved by a novel iterated local search (ILS) metaheuristic. A mixed integer programming (MIP) model from the literature is used to validate the proposed method in 32 grid networks with sizes 6x6, 10x10, 20x20 and 30x30, with two different number of fire suppression resources (64 problems). Our ILS produced optimal solutions in 40 cases out of 41 known optimal lower bounds. The proposed method's effectiveness is also due to its short computing times and small coefficients of variation of the objective function values.

We also provide a categorised literature review on fire suppression deterministic optimisation models, from which we conclude that approximate collaborative approaches seldom have been applied in the past and, according to the results obtained, can successfully address the complexity of fire suppression, reaching good quality solutions even for large scale instances.

1. Introduction

Wildfires have a major impact on human and environmental life. In UNDRR (2019), it is reported that wildfires caused 71 deaths and affected more than 19 thousand people, on average, per year, between 2000 and 2017. Solely in 2018, wildfire caused 247 deaths and affected more than 250 thousand people. Recent wildfires with major impact include the 2020 California wildfires, with a size of more than 1700 thousand ha, that provoked 32 direct deaths; the 2019–2020 Australia wildfires with a size of more than 1800 thousand ha, that provoked 34 direct deaths and 445 deaths by smoke inhalation; the 2019 Amazon wildfires with a size close to one thousand ha; and the 2017 (June and October) Portugal wildfires that caused 115 direct deaths[1].

In the aforementioned countries and in many others, large amounts of resources are allocated to prevent and suppress wildfires in order to attempt to mitigate their negative impacts. These include, of course, the direct threat to human lives but also, through the emission of gas and particles, health effects and, at a different scope, global temperature increase[2]. Through the destruction of forest resources and wildlife habitats, economical and environmental impacts are also major consequences that must be considered.

Operational research and optimisation have been used in addressing forest fires since the 1960s. For example, as early as 1963, a model for determining the fire-suppression force that minimises the total cost of an initial attack was proposed (Jewell, 1963).

In Martell (1982), a review of operational research studies in forest fire management is conducted. Although some of the reviewed areas had significant evolutions since the early 1980s, e.g. fire detection, the core of other problems remains the same, as well as the potential of operational research to contribute to their mitigation. Examples are fire load management (prevention

* Corresponding author.
E-mail address: andbergs@usp.br (A.B. Mendes).

[1] Figures from https://www.en.wikipedia.org/wiki/List_of_wildfires, accessed 2020-11-1.

[2] https://www.public.wmo.int/en/media/news/widespread-fires-harm-global-climateenvironment, accessed 2021-01-04.

https://doi.org/10.1016/j.ejor.2022.04.037
0377-2217/

planning and fuel management, surveyed more recently in Chung (2015)) and fire suppression, which activities, in the same reference, are divided in resource acquisition and strategic deployment, resource mobilization, initial attack dispatching and extended attack management.

A more recent overview of operational research methods and applications in forest fire management are Minas, Hearne, & Handmer (2012) and Martell (2015). In the latter, the author refers that “many of the large fire management challenges that Shephard & Jewell (1961) initiated research on the application of OR/MS to fire management remain. That being said, it has not been for lack of effort.”

The relevance of fire suppression resource management, in which operational research and optimisation, we believe, play a decisive role, is supported, for example, in Fernandes, Pacheco, Almeida, & Claro (2016), where extremely large fires in Portugal from 2003 to 2013 were analysed to conclude that more effective identification and exploration of containment opportunities (i.e. resources management) are preferable to higher fire-suppression resourcing.

In Duff & Tolhurst (2015) fire suppression activities are divided in two groups: one related to preparedness and the other with response. Preparedness activities are further divided in four levels: protection analysis, resource location-allocation, readiness and detection. Similarly, response activities are further divided in three levels: dispatch, travel and suppression works. The authors mention that fire suppression fits in the preparedness and response components of the management of emergencies. Preparedness is related with the activities prior to an ignition and response to the ones after. Overviews on the issues and approaches for large-fire management can be found in Dunn, Thompson, & Calkin (2017) and Thompson, Silva, Calkin, & Hand (2017).

In this paper, we focus on decisions related to the resources available to fire suppression. Examples of these resources are fire crews and their equipment, airtankers and fire trucks. We model their effect in a given location as the delay in the fire spread they provoke, which allows to model direct extinguishment of the flames and fireline-based containment. The proposed approach integrates the spatial and time dimensions in fire spread and also in the resources usage in order to minimise the burned area of a landscape given an ignition point and fire travel times between adjacent cells. The proposed approach is flexible enough to model the use of resources in an initial attack or the protection of assets in large fire management, meaning that other objectives than minimising the burned area in a given time horizon may be considered.

This problem was previously addressed by Alvelos (2018), who proposed several MIP models to solve different variants. The author solved small landscapes (6x6), aiming to demonstrate the models' applicability. In this work, we propose a novel iterated local search (ILS) metaheuristic and demonstrate its effectiveness by solving a set of 64 test cases for different landscapes, including real-sized instances. For the 48 smaller problems, with up to 400 nodes, the solutions provided by the ILS can be compared with the bounds provided by the MIP model, while for the larger instances, the model from Alvelos (2018) can not be solved due to memory limitations.

The sequence of this paper includes a literature review in Section 2, followed by the problem definition and a MIP model in Section 3. The proposed metaheuristic is described in Section 4, and dummyTXdummy- the computational results are presented and commented in Section 5. Conclusions are drawn in Section 6.

2. Literature review

Several optimisation models, in particular mixed integer programming (MIP) models, have been proposed to address fire suppression problems. We conduct a categorised literature review based on the interaction of the fire representation and optimisation model. Three types of approaches are considered: sequential, integrative and collaborative. We focus on deterministic models, in concordance with our approach.

For stochastic models, we refer the interested reader to the stochastic integer programming approaches of Haight & Fried (2007) and Belval, Wei, & Bevers (2019) and references therein.

2.1. Sequential

We first review models derived from well-known MIP formulations, such as location, allocation and routing, and, in some cases, their integration. These models are adapted to the fire preparedness and/or response through the use of parameters that represent fire suppression requirements that must be met by the activity of the resources. We first consider examples where fire suppression requirements are modelled as demands and then as a fireline length that must be built to contain fire.

The fist step in addressing a fire suppression optimisation problem with one of the these approaches is to characterize fire through the parameters of the model (of course also defining other parameters, for example related to resources) and, in a second step to obtain a solution by solving the model. Accordingly, we name this type of approach as sequential.

2.1.1. Demand-based

In covering models (Dimopoulou & Giannikos (2001); Marianov & ReVelle (1992)), decisions are related to where to locate resources (e.g. vehicles, truck stations) to cover a set of demand points (e.g. regions) representing potential fire events. In allocation models (MacLellan & Martell (1996); Mees & Strauss (1992); Mees, Strauss, & Chase (1993)) decisions are related to which combination of resources (e.g. airtankers) should be allocated to each demand (e.g. fire segment, initial attack base). In vehicle routing models (van der Merwe, Minas, Ozlen, & Hearne (2015); Roozbeh, Ozlen, & Hearne (2018); Wu, Cheng, & Feng (2019)) decisions are related to the definition of a sequence in which a set of vehicles (e.g. tankers) should visit a set of assets, within given time windows, in order to maximise the value protected. Both spatial and time dimensions are considered for the resources but they do not interact with fire spread which is modelled by parameters.

2.1.2. Fireline-based

In fireline-based models, fire is represented by a pre-defined perimeter at different time instants. A fire is extinguished when the fireline built by the resources is greater than or equal to the fire perimeter (perimeter condition). In Wiitala (1999) and Donovan & Rideout (2003) dispatch decisions are addressed: which resources send to a fire site to minimise the cost of fire containment. In Kirsch & Rideout (2005); Rideout, Wei, & Kirsch (2011); Rodríguez-Veiga, Ginzo-Villamayor, & Casas-Méndez (2018) MIP models are derived for deciding the allocation, and its duration, of resources to fires. In Zambon, de Rezende, & de Souza (2018), decisions are related to which barriers (from the ones enumerated in a preprocess step) to built and in which sequence to maximise the salvaged area defined by the faces that are protected.

2.2. Integrative

In integrative approaches, a single model includes optimisation and fire spread: decisions on resources affect fire spread and vice-versa. In scheduling models, the sequence in which fires (or fire segments) are visited influence the time required for their suppression and the time for suppression of fires influence the sequence. In models where adjacency relations determine the fire spread, the location of resources changes the adjacency relations which changes the optimal location of resources. Lastly, in models based on the minimum travel time principle, resource locations influence the fire travel times to adjacent nodes which influence the optimal location of resources.

2.2.1. Scheduling

In Rachaniotis & Pappis (2006), Pappis & Rachaniotis (2010) and Rachaniotis & Pappis (2011) fire suppression is modelled as a scheduling problem where fires correspond to jobs to be processed by a single resource. Setup times (sequence dependent) correspond to the resource movements between fires. Fire spread is modelled as the time needed for suppression and increases with the time from ignition according to a Rothermel empirical model (based on the wind speed, type of fuel and fuel loading).

2.2.2. Adjacency

In Belval, Wei, & Bevers (2015), the decisions, modelled in a MIP model, are related to the positioning of resources in a network representing the landscape. An ignited cell spreads fire to all its neighbour cells that are flammable and do not have a resource. A more theoretical problem that relies on modelling fire spread through adjacency is the firefighter problem (Blum, Blesa, García-Martínez, Rodríguez, & Lozano (2014); Develin & Hartke (2007); Finbow & MacGillivray (2009); Hu, Windbichler, & Raidl (2015); Michalak (2014, 2017); Ramos, de Souza, & de Rezende (2020)). This problem is defined on a graph where fire spread is simulated by a sequence of time steps. An ignition occurs in one vertex at the first step. In every step, vertices adjacent to a burned vertex also burns, except for the ones where a resource was located. At the beginning of each step a (fixed) number of resources is available. The objective is to minimise the number of burned vertices.

In Wei et al. (2021); Wei, Thompson, Haas, Dillon, & O'Connor (2018); Wei, Thompson, Scott, O'Connor, & Dunn (2019) models based on the definition of potential wildfire operational delineations (POD, polygons whose boundaries may facilitate fire control operations, such as roads and fuel transitions) is presented. Fire spread is modelled through adjacency between POD. Decisions include the definition of the boundaries and protection points where the (limited) resources should work.

2.2.3. Minimum travel time

In integrative approaches based on the minimum travel time (MTT) principle Finney (2002), the landscape is represented by a network (with any topology and, theoretically, any resolution). Nodes represent locations (e.g. stands or cells). Arcs represents adjacency between locations. The minimum travel time (MTT) principle states that the fire arrival time at a node is the shortest path from the ignition node to that node with respect to the fire travel times between all adjacent nodes. In a MIP integrating the MTT and optimisation, the fire spread is modelled with decision variables related to fire arrival times (and to the shortest paths from the ignition to the cells) and decision variables related to the location of resources. Fire spread and resources interaction are modelled by the increase of the fire travel times from a node where a resource is located to adjacent nodes.

To our best knowledge, MTT fire spread and optimisation has been first integrated in Hof, Omi, Bevers, & Laven (2000) through a MIP for maximising the fire arrival times at cells that must be protected. In the MIP model of Wei, Rideout, & Hall (2011), the objective is to minimise the value of the cells that burn within a time horizon. In Alvelos (2018), besides these two objectives, objectives related to fire containment (by perimeter and by fire inactivity), are also considered. In this latter reference, two noticed issues of the previous models were overcome.

The first one is that, as noted in both works, with a straightforward model, arrival times in cells not belonging to the fire paths may not comply with the shortest ones. In Wei et al. (2011) this issue is addressed by a sequential procedure involving the solution of two MIP models. In Alvelos (2018), linear programming optimality conditions are used to derive a single MIP model which, independently of the objective function, always provides the correct arrival times at every cell.

A second issue is related with the time availability of the resources from the start. A possibility to incorporate that resources are not immediately available is to forbid cells close to the ignition to receive resources, but this imply that the fire arrival times at those cells is known. In Wei et al. (2011), an approximate iterative procedure, based on the MIP model, is presented to address the multi-period problem, where a set of resources is made available at the beginning of each time period, overcoming that issue. In Alvelos (2018), this issue is addressed by defining time instants where the resources become available, treating time both as continuous (for fire arrival times at cells) and discrete (instants when resources become available/are located).

2.3. Collaborative

In collaborative approaches, two modules (optimisation and fire spread) exchange information. Typically, the optimisation module provides solutions (e.g. location of resources) and receives their evaluation (e.g. burned area). Collaborative approaches have a big potential in fire suppression as they accommodate, virtually, any fire spread model (from physical to empirical) and, virtually, any search method (e.g. meta-heuristic).

In Chi et al. (2003) fire propagation is modelled by a cellular automata (with deterministic rules for state transitions) which evaluates different resources usage from the search space of a genetic algorithm. A genetic algorithm is also proposed in HomChaudhuri, Kumar, & Cohen (2013) where the fitness of each individual is obtained by simulating the fire spread taking into account the resources location and the fireline construction rate.

2.4. Comparison and contributions

We now briefly compare the three different types of approaches and describe what we think are the major contributions of this paper.

Sequential approaches are very limited in terms of fire spread modelling. Integrative approaches may incorporate more detailed fire spread models but are difficult to solve given their complexity and size (in case of MIP models even with state-of-the-art solvers).

Collaborative approaches allow, virtually, any fire spread model to be used for evaluating solutions generated or modified by an optimisation module. These approaches can be seen as particular cases of the general scheme of simulation-optimisation, which has been applied successfully in many areas (Figueira & Almada-Lobo, 2014).

A first main contribution of this paper is therefore to highlight the advantages of addressing large scale wildfire optimisation problems using collaborative approaches, paving the way for their use with other optimisation methods / fire spread models in addressing wildfire problems.

The second main contribution is the design and computational validation of a collaborative approach that combines a metaheuristic with a MTT fire spread model solved by Dijkstra's algorithm and provides good quality solutions (proved to be optimal ones in most cases) for large instances of a relevant wildfire suppression problem.

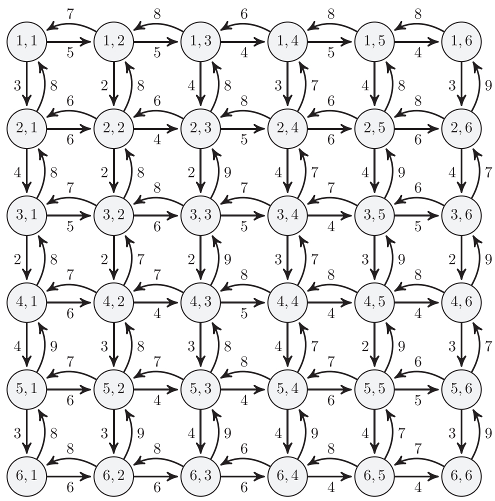

Fig. 1. Example of a network representing a landscape and fire travel times (Alvelos, 2018).

3. Problem definition and modelling

Let (N, A) be a graph representing a rasterised landscape. The set of nodes N contains cells of the landscape, while the set of arcs A represents adjacency relations between them (i.e. direct fire transmission is possible), as exemplified in Fig. 1. It is assumed that the fire ignition takes place in a single node represented by $ign \in N$.

The fire rate of spread between nodes depends on several issues, such as the amount of fuel at each node (e.g. flammable vegetation), the wind direction and intensity, and the terrain slope. It is a premise that each cell contains a homogeneous area with respect to weather, topography, and fuel. The fire spread time between adjacent nodes i and j is given by c_{ij}, which can be estimated by fire propagation simulators such as Finney (1998).

We consider a set of K time instants, $K = \{b_1, b_2, \ldots, b_h\}$, with h being the last instant of the time period under consideration, and a set of fire fighting resources R. In the k^{th} instant, time b_k, a given number resources, a_k, becomes available and can be located in the unburned nodes of the network at this instant or later. Locating a resource in a node implies that the node will not burn and the fire spread to adjacent unburned nodes is delayed by a known value Δ.

For a fire ignited in instant 0, at node $ign \in N$, the problem consists of determining when and where to position the available resources so that the total burned area is minimised. A secondary objective is to minimise the total number of deployed resources. This approach resorts to lexicographic optimisation where each solution with a smaller burned area is better independently of the number of resources; the number of resources is only used to differentiate solutions with the same burned area. In a rasterised landscape, all cells have the same area, and the minimisation of the number of burned nodes thus produces the same effect. By adopting $\epsilon = 1/|R|$ in the objective function, a single model can be used if such weight does not pose numerical difficulties to the solver. If that is the case, two models can be solved sequentially, the first for the burned area objective (or the number of burned nodes) and the second, with an additional constraint fixing the optimal burned nodes, for the number of resources.

The mixed integer programming (MIP) model proposed by Alvelos (2018) is presented to make this paper self-contained. Moreover, it is intended to assess the maximum landscape size that commercial solvers can solve and use to model to evaluate the performance of the proposed heuristics. The model relies on the following sets and parameters: N - set of nodes (indices i, j); A - set of arcs; R - set of resources (index r); K - set of time instants (indices k, g); h - target instant, in which the solution (burned area and number of deployed resources) are minimised ($h \in K$); ign - ignition node ($ign \in N$); n - number of nodes ($n = |N|$); c_{ij} - fire spread time between the (center of) node i and the (center of) node j in that direction; $c^{\max}$ - maximum fire spread time between any two nodes; a_k - number of resources that become available at instant k (b_k); Δ - delay, expressed in time units, of the fire arrival to an unburned adjacent node to the one that received a resource; ϵ - weight of the total number of resources in the objective function ($\epsilon = 1/|R|$).

The decision variables present in the model are: x_{ij} - the number of shortest paths (each one beginning in the root and ending in a different node) that include arc ij; t_i - length of a shortest path between the root and each node i (a value that may vary according to the resources that have been installed as they modify the fire propagation paths); s_{ij} - slack variable that is zero whenever arc ij belongs to a shortest path; q_{ij} - a binary variable that equal 1 if arc ij belongs to a shortest path, and 0 otherwise; y_i^k - a binary variable that equals 1 if node i is burned at instant k (b_k), and 0 otherwise; o_k - number of resources available but not used at instant k and therefore available at instant $k+1$; z_i^{kr} - a binary variable that equals 1 if node i receives resource r at instant k (b_k), and 0 otherwise.

$$\text{Minimise} \sum_{i \in N} y_i^h + \epsilon \sum_{i \in N} \sum_{k \in K} \sum_{r \in R} z_i^{kr} \tag{1}$$

Subject to:

$$\sum_{ign, j \in A} x_{ign,j} = n - 1, \tag{2}$$

$$-\sum_{ij \in A} x_{ij} + \sum_{ji \in A} x_{ji} = 1, \quad \forall i \in N \setminus \{ign\} \tag{3}$$

$$t_{ign} = 0 \tag{4}$$

$$x_{ij} \leq (n-1) q_{ij}, \quad \forall ij \in A \tag{5}$$

$$\sum_{i \in N} \sum_{k \in K} z_i^{kr} \leq 1, \quad \forall r \in R \tag{6}$$

$$\sum_{r \in R} \sum_{k \in K} z_i^{kr} \leq 1, \quad \forall i \in N \tag{7}$$

$$\sum_{i \in N} \sum_{r \in R} z_i^{1r} + o_1 = a_1, \tag{8}$$

$$\sum_{i \in N} \sum_{r \in R} z_i^{kr} + o_k = a_k + o_{k-1}, \quad k = 2, \cdots, |K| \tag{9}$$

$$z_i^{kr} \leq 1 + (t_i - b_k)/b_k, \quad \forall i \in N, \forall k \in K, \forall r \in R \tag{10}$$

$$t_j - t_i + s_{ij} = c_{ij} + \Delta \sum_{r \in R} \sum_{k \in K} z_i^{kr}, \quad \forall ij \in A \tag{11}$$

$$s_{ij} \leq ((n-1)c^{max} + (|R| - 1)\Delta)(1 - q_{ij}), \quad \forall ij \in A \tag{12}$$

$$y_i^k \geq \frac{b_k - t_i + 1}{b_k} - \sum_{r \in R} \sum_{l=1,\ldots,k} z_i^{lr}, \quad \forall i \in N \setminus \{ign\}, \forall k \in K \tag{13}$$

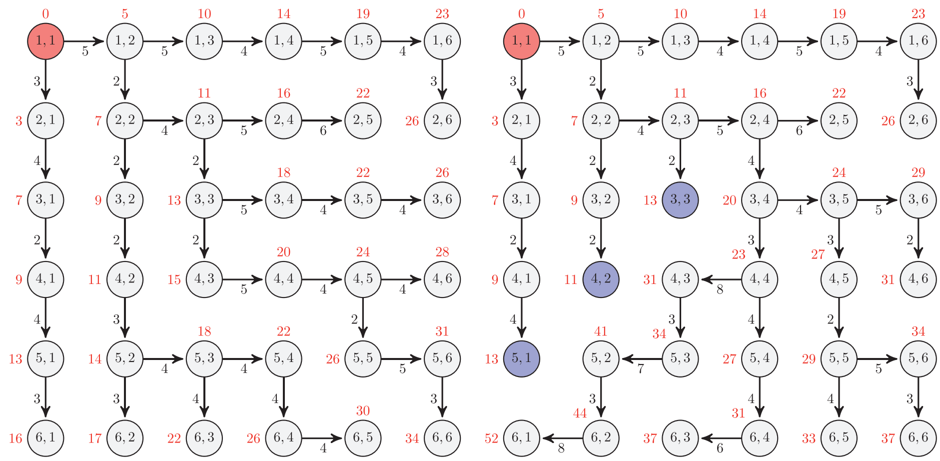

Fig. 2. Fire shortest paths and arrival instants at all nodes for ignition at node (1,1) (left); Fire shortest paths and arrival instants at all nodes after placing resources at (3,3), (4,2) and (5,1) (right).

$$y_i^k \le 1 + \frac{b_k(1 - \sum_{r\in R}\sum_{l=1,\dots,k} z_i^{lr}) - t_i}{(n-1)c^{max} + (|R|-1)\Delta}, \quad \forall i \in N, \forall k \in K \tag{14}$$

$$o_k \ge 0, \quad \forall k \in K \tag{15}$$

$$t_i \ge 0, \quad \forall i \in N \setminus \{ign\} \tag{16}$$

$$x_{ij}, s_{ij} \ge 0, q_{ij} \in \{0, 1\}, \quad \forall ij \in A \tag{17}$$

$$y_i^k \in \{0, 1\}, \quad \forall i \in N, \forall k \in K \tag{18}$$

$$z_i^{kr} \in \{0, 1\}, \quad \forall i \in N, \forall k \in K, \forall r \in R \tag{19}$$

In the this model, the objective function (1) minimises the total number of burned nodes at the target instant plus a weighted number of assigned resources. Constraint (2) forces that $n-1$ paths departure the ignition node *ign*, while constraint (3) guarantees that one path reaches each node in the network. Constraint (4) states that the fire arrival time of ignition node is zero. Constraint (5) activates the binary variable q_{ij} if arc (i, j) belongs to a shortest path. Constraint (6) states that a resource can be assigned at most once to any node. Analogously, constraint (7) guarantees that each node can receive at most one resource throughout the planning period. Constraint (8) allows assigning resources at instant 1, based on the number of resources that were released (a_1). Constraint (9) controls the number of available resources by balancing the number of unassigned resources at the end of each time period. In constraint (10), it is checked whether node i is burned at instant a_k. This happens when the fire arrival time t_i is less than the evaluated instant. In such cases, node i cannot receive a resource. On the other hand, if the fire arrival instant is greater than or equal to a_k, then node i can receive a resource at this instant. Constraint (11) calculates the fire arrival instant at node j, having node i as origin. In case a resource has been assigned to node i, a delay in the fire arrival at node j is guaranteed by parameter Δ. Constraint (12) forces a slack variable s_{ij} to be zero whenever arc (i, j) belongs to a shortest path (i.e., when q_{ij} is one). Constraints (13) and (14) define if node i is burned at instant k. These constraints, together with constraint (10), allow node i to receive a resource at the same instant of the fire arrival at the node. Finally, constraints (15) to (19) define the variables' domain.

To illustrate the fire propagation model used, the left part of Fig. 2 presents the fire shortest paths to all 35 nodes when the ignition takes place at node (1,1), for the example shown in Fig. 1. The values marked in red are the fire arrival instants at the nodes calculated by Dijkstra's shortest path algorithm when no action is taken to protect them, i.e. no resources are used.

In right part of Fig. 2 we represent a solution where three resources are located in nodes (3,3), (4,2) and (5,1). Note that for the solution to be feasible, the fire arrival time at those nodes must be greater than the instant they become available (i.e. the resources must be available before 13, 11 and 13 time units, respectively).

Because of the delay (assuming Δ is large enough) provoked by locating the resources, the fire paths of the previous successors of (3,3), (4,2) and (5,1), do not include them any longer. For example, node (3,4), whose fire arrival instant was 18, has a new shortest path reaching it from node (2,4) with an increased fire arrival instant equal to 20. It is interesting to note how the modified fire arrival instants varies throughout the network. In node (5,2), for example, the fire arrival increased from 14 to 41, as node (4,2) became protected and a longer fire path is needed to reach it. By considering the time horizon as 20, the number of burned nodes is 22 with no resources and 13 with the three resources.

4. Solution method

An iterated local search (ILS) algorithm is proposed to minimise the total number of burned nodes at the given target instant h plus the weighted number of resources. The ILS metaheuristic, as described in Lourenço, Martin, & Stützle (2003), follows the general structure presented in Algorithm 1. In line 1, an initial solution s_0

Algorithm 1: Iterated Local Search.

```
1 (s0, f0) ← MultiStartConstructiveHeuristic()
2 (s*, f*) ← LocalSearch(s0)
3 while Stopping criterion is not met do
4 |   (s', f') ← Perturbation(s*)
5 |   (s*', f*') ← LocalSearch(s')
6 |   s* ← AcceptanceCriterion(s*, s*')
7 return s* and f*.
```

is generated by a multi-start constructive heuristic with objective function value f_0. This solution is then submitted to a local search procedure in line 2, returning solution s^* with objective function value is f^*. ILS' main loop takes place in lines 4 to 6, while the stopping criterion, defined as the maximum number of iterations without improvement, is not met. In line 4, the s^* solution is perturbed, resulting in solution s' with objective function value f'. This solution is submitted to a local search procedure in line 5, returning solution $s^{*\prime}$ with objective function value $f^{*\prime}$. Finally, the acceptance criterion is applied in line 6, whereby only improved solutions are admitted.

As described, ILS relies on searching a solution space by iteratively exploring the neighborhood of a current solution and moving to a more distant solution when no improvement can be made. ILS is one of the most well-known single-solution meta-heuristics (Gendreau, Potvin et al., 2010). These are characterized by a compromise between fast computational running times (usually much faster than population-based methods as, for example, genetic algorithms) and faster high quality-solutions (usually much better than constructive algorithms). The main routines are detailed in the next subsections.

4.1. Constructive heuristic

The constructive heuristic is a multi-start probabilistic procedure whose structure follows the concepts discussed in Martí, Resende, & Ribeiro (2013). Each iteration of Algorithm 2 builds a randomised solution s according to the general procedure described in Algorithm 3 , which is kept as the best initial solution s^* if the objective function value f^* improves.

Algorithm 2: MultiStartConstructiveHeuristic().

$f^* \leftarrow \infty$
while Maximum iterations has not been reached **do**
  $s = $ ConstructRandomSolution()
  if $f(s) < f^*$ **then**
    $s^* \leftarrow s$
    $f^* \leftarrow f(s)$
return s^* and f^*.

Algorithm 3: ConstructRandomSolution().

1 **while** *there are unburned nodes with fire arrival instants less than h* ***and*** *there are resources available* **do**
2   Find the least release time of a resource.
3   Run Dijkstra's algorithm to determine the fire arrival instants.
4   Identify the set of burned nodes.
5   Sort nodes in ascending order by their fire arrival instants.
6   Build a restricted candidate list of unburned nodes.
7   Randomly select a node from the candidate list.
8   Assign a resource to the chosen node.
9   Update the fire travel time to its adjacent unburned nodes.
10 **return** solution and its objective function value.

In each iteration of the constructive heuristic, Dijkstra's shortest path tree algorithm is used to determine the fire propagation paths and the corresponding fire arrival instant at each node. Based on those times, one resource is located in one of the locations of a restricted candidate list made of unburned locations. When a resource is placed on an unburned node, it becomes protected and delays the fire propagation path that was meant to go through it. Hence, the heuristic selects promising nodes to place the resources, considering the resources' release instants and the fire arrival times at the nodes.

As presented in Algorithm 3, the randomised construction of a solution is an iterative procedure with two stopping rules (line 1): the algorithm is interrupted if all unburned nodes have their fire arrival instants greater than h (since the solution is evaluated at instant h, there is no need to continue) or if no further resources are available (there is nothing to be done). If the stopping criterion is not met, the procedure will first find out the least release instant of a resource (line 2). This is important as no action can be taken before this instant. The fire arrival time to all nodes is then calculated with Dijkstra's shortest path algorithm (line 3). The nodes whose fire arrival instants are inferior to the least release instant of a resource cannot be protected and are set as burned (line 4). The remaining unburned nodes are sorted in the ascending order of their fire arrival instants (line 5), and a restricted candidate list is built (line 6) containing the unburned nodes that will first burn if no resources are deployed before the fire arrival. One node is randomly selected from the restricted candidate list (line 7), and a resource is assigned to it (line 8). As a consequence, the unburned adjacent nodes have the fire propagation times delayed by parameter Δ (line 9).

4.2. Local search

The general structure of the local search is described in Algorithm 4 and consists of removing the resource from each node, one at a time, and evaluating the impact of placing the resource in the neighbouring nodes to those with resources. By this we mean adjacent nodes as well as diagonal nodes (e.g. node (1,4) is diagonal to (2,5)). This type of movement aims to form fire suppression barriers by having fire suppression resources positioned adjacently - later, on the perturbation phase, we also include nodes

Algorithm 4: LocalSearch().

repeat
  for *each* $node_0$ *with a resource* **do**
    Remove the resource from $node_0$.
    Update the fire propagation time to the unburned adjacent nodes of $node_0$.
    Run Dijkstra's algorithm to determine the fire arrival instants.
    neighbourhood $\leftarrow$ GenerateNeighbourhood($node_0$)
    for *each* $node_1$ *in neighbourhood* **do**
      Place the available resource at $node_1$.
      Update the fire travel time to the adjacent unburned nodes of $node_1$.
      Run Dijkstra's algorithm to determine the fire arrival instants.
      Determine the solution feasibility.
      if *the solution is feasible and the best possible improvement* **then**
        Save movement as the *bestMovement*.
      Remove resource from $node_1$.
      Update the fire travel time to the adjacent unburned nodes of $node_1$.
    Restore the resource to $node_0$.
    Update the fire propagation time to the unburned adjacent nodes of $node_0$.
  if *there is improvement* **then**
    Execute the *bestMovement*.
until *no improvement*;
return solution and its objective function value.

that not necessarily form a barrier. It is essential to highlight that more than one barrier is admitted.

The complete list of the neighbouring nodes to those with resources is called an extended neighbourhood. As the size of this neighbourhood can be very large, a reduction scheme is applied. The first nodes to eliminate are the burned nodes. The remaining nodes are sorted in ascending order of their fire arrival instants, and a reduced neighbourhood is built with the first elements of this list (more details are given in the computational results section).

The local search procedure starts by removing the resource from one node (line 3). Then, the fire propagation time to the adjacent unburned nodes is updated (line 4), which means that the delay in propagating the fire (parameter Δ) is eliminated. In the sequence, the fire arrival instants to all nodes are determined by Dijkstra's shortest path algorithm (line 5). The next step consists of defining a reduced neighbourhood for placing the resource (line 6). Then, it is assessed the impact of placing the resource at each node belonging to this neighbourhood (lines 7 to 17).

This evaluation begins by placing the resource at one node of the neighbourhood (line 8) and modifying the corresponding fire propagation time to its unburned adjacent nodes (line 9). In the sequence, Dijkstra's shortest path algorithm is called (line 10) to determine the fire arrival instants to all the nodes. Before evaluating the number of burned nodes, a feasibility check has to be made (line 11). As the resource was removed from its original position (line 3) and reinserted in a different node (line 8), the fire propagation paths to all the nodes are affected, and the fire arrival instants at all nodes are likely to change. One has to be aware that a node with a resource cannot have its new fire arrival time inferior to the instant the resource was previously assigned to it. Such a situation is inconsistent and characterizes an infeasible move. In case the movement is feasible, and this modification turns out to be the best possible improvement, it is saved (lines 12 and 13) for future modification (line 18). In lines 15 and 16, the resource is removed from $node_1$, and the fire propagation time between its adjacent nodes is updated. When all nodes belonging to the neighbourhood have been examined, the resource is restored to $node_0$ (line 18), and the fire propagation time to its unburned adjacent nodes are updated (line 19). Finally, in line 21, after the whole neighbourhood of all nodes with resources has been evaluated, and if it is possible to improve the solution, the best movement is executed (line 22). Otherwise, the procedure is halted by the outer cycle condition (line 24).

4.3. Perturbation scheme

The perturbation scheme is meant to generate solutions that are not too close to the incumbent solution and neither too distant, enabling to escape local optima, hopefully keeping good-quality features of the solution. In this regard, three types of perturbations were implemented and are described in Algorithm 5 . The first one is called with probability $prob_1$ and aims to reduce the number of assigned resources (lines 4 to 6). The second type of perturbation is called with an accumulated probability of $prob_1 + prob_2$ and adds a resource to the solution (lines 9 to 17). This perturbation is only called when there is at least one available resource; otherwise, $prob_2$ is set to 0 (line 1). The third perturbation forces that a predetermined number of resources be moved from their current positions to other nodes, provided that the solution remains feasible.

The first type of perturbation can improve the solution in situations where not all resources are needed to minimise the total burned area at the target instant h. This may happen when too many resources are available or the fire travel times are large with respect to target time. In those cases it is important to minimise

Algorithm 5: Perturbation().

```
1  Set prob_2 to 0 if there are no available resources.
2  Rnd = random()
3  if Rnd < prob_1 then
4      Remove a resource from the node with the largest
       deployment instant.
5      Update the resource availability.
6      Update the fire travel time to its adjacent nodes.
7  else
8      if Rnd < prob_1 + prob_2 then
9          Find the least release time of an available resource.
10         Run Dijkstra's algorithm to determine the fire arrival
           instants.
11         Identify the set of candidate nodes to receive the
           resource.
12         Sort candidate nodes in ascending order by their fire
           arrival instants.
13         Build a candidate list for receiving the resource.
14         Randomly select a node from the candidate list.
15         Assign a resource to the chosen node.
16         Update the resource availability.
17         Update the fire travel time to its adjacent nodes.
18     else
19         mod = 0
20         failure = 0
21         while mod < maxModidications and
           failure < maxFailures do
22             Randomly select a node with resource.
23             Remove the resource.
24             Update the fire propagation time to its adjacent
               nodes.
25             Run Dijkstra's algorithm to determine the fire
               arrival instants.
26             Generate the neighbourhood.
27             Randomly select a node from the neighbourhood.
28             Place the resource at the selected node.
29             Update the fire propagation time to its adjacent
               nodes.
30             Run Dijkstra's algorithm to determine the fire
               arrival instants.
31             Assess the solution feasibility.
32             if the solution is feasible then
33                 mod = mod + 1.
34                 failure = 0.
35             else
36                 Undo the proposed movement.
37                 failure = failure + 1.
38 Run Dijkstra's algorithm to determine the fire arrival instants.
39 return solution and its objective function value.
```

the number of resources used, keeping the burned area at the minimum.

It must be observed that the constructive heuristic initially defines the number of resources, and the subsequent local search does not modify this quantity. In cases where the initial solution was built using more resources than what is actually needed, this type of perturbation allows adjusting the solution, keeping the burned area unchanged. The idea is simple and consists of finding the node whose resource deployment instant is the largest (algorithm 5, line 4) and remove the resource. Lines 5 and 6 are necessary to update the resource availability and update the fire travel time to the adjacent nodes.

The second type of perturbation consists of adding a resource to the solution, if available. This perturbation, however, is called less frequently, as it is not expected that a solution will demand more resources than it was defined in the constructive phase.

The process is very similar to the steps described in algorithm 3 and consists of finding the least release time of an available resource (algorithm 5, line 9) and call Dijkstra's shortest path algorithm to determine the fire arrival instants in the whole network (line 10). A list of candidate nodes is elaborated, including all nodes whose fire arrival instants are greater than or equal to the instant that the resource is released and do not have a resource (line 11). This list of candidate nodes is sorted in ascending order by their fire arrival instants (line 12), and a restricted candidate list is built (line 13). A node is randomly selected from this list (line 14) to receive a resource (line 15). The resource availability is updated (line 16), and the fire travel time to its adjacent nodes is updated (line 17).

The third type of perturbation is most frequently called (with a probability of $1 - prob_1 - prob_2$) and consists of modifying a predefined number of resources from their positions (lines 22 to 39). While the maximum number of modifications has not been reached and the number of failures has not reached its maximum value, a resource is moved from its current node. First, a node with a resource is randomly selected (line 22), and the corresponding resource is removed (line 23). Consequently, the fire propagation time to its adjacent nodes must be updated (line 24), and Dijkstra's shortest path algorithm is called (line 25) to determine the fire arrival time at all nodes after the resource was removed. The neighbourhood for placing the resource is generated (line 26) in the same way as in the local search procedure, but without considering that a node must be adjacent or diagonal to a node with a resource. This means that any unburned node can be part of the neighbourhood, thus generating a broader list of candidate nodes. However, as the neighbourhood has a maximum allowed size (further defined), the candidate nodes are sorted in ascending order by their fire arrival instants, and only the nodes with the smaller instants are considered. One node from this neighbourhood is selected (line 27) to receive a resource (line 28).

Dijkstra's shortest path algorithm is called (line 30) to determine the fire arrival instants to assess the solution feasibility (line 31). This verification is needed to check if the new fire arrival instants at the nodes with resources are inferior to the instants that the resources were released. If this is the case, we have an infeasible solution. Otherwise, the counter *mod* is incremented, and the *failure* counter is set to zero (lines 33 and 34). If the solution is infeasible, the movement made in lines 23 and 28 must be undone (line 36), and the *failure* counter is incremented (line 37). Before returning the solution (line 43), the objective function has to be updated, and this is done by first calling Dijkstra's shortest path algorithm for determining the fire arrival instants at all nodes, which allows assessing the resulting number of burned nodes.

5. Computational results

5.1. Instances and ILS calibration

Computational tests were made with the mixed integer programming model and the iterated local search metaheuristic. Instances were created with four different grid sizes: 6x6, 10x10, 20x20 and 30x30, as in Hof et al. (2000), Minas, Hearne, & Martell (2014) and Belval et al. (2015). In all cases the ignition takes place in a quasi-central node, i.e. nodes (3,3), (5,5), (10,10) and (15,15) for grid sizes 6x6, 10x10, 20x20 and 30x30, respectively, thus allowing the fire to propagate in all directions. Besides the grid size and the ignition node, one needs to know the estimated fire spread time between any pair of adjacent nodes. Several aspects influence the fire propagation spreading rate and define the propagation time more precisely (Finney, 1998). The estimate of the fire propagation times, however, is beyond this paper's scope.

To generate the test instances, uniform distributions are used to define the spreading times (in minutes) between adjacent nodes in the directions indicated in Table 1, considering the wind as the main influence. For example, in instance one, the fire propagation times southward and eastward are drawn from the uniform distributions U(2,4) and U(4,6), respectively. These distributions have the least minimum and maximum values among the four directions, indicating that the resulting wind direction is southeast. For each grid size, eight variations are proposed with different time propagation distributions (32 problems); instance 1 was extracted from Alvelos (2018) and is represented in Fig. 1.

The instances are evaluated for two different number of resources. The first set considers that two resources become available at instant 10 and three resources become available at instant 15. The second set considers that three resources become available at instants 10 and 15. In Table 1, two additional pieces of information on the instances are given in the last two columns. First, in the 'burned nodes' column, one will find the number of nodes whose fire arrival times are inferior to when resources are first released (instant 10) and will inevitably burn. Second, the 'candidates first resources' column indicates the number of nodes that are candidates to receive the resources released at instant 10, including all nodes whose fire arrival instants are between 10 and 14. Intuitively, the greater the number of candidate nodes relative to the number of resources, the more difficult the instance is. The fire delay implied by locating a resource, Δ, is assumed to be 50. For all cases, the objective function is evaluated at instant $h = 28$ minutes, when the number of burned nodes is assessed, and decisions on other actions can be taken, such as requesting additional resources or planning an extended attack (Martell, 2015).

The proposed heuristic requires the calibration of general parameters (ILS) and others relative to the constructive heuristic (CH), local search (LS) and the perturbation scheme (Pert). The adopted values are presented and if not otherwise indicated are common to all grid sizes - *maximum number of perturbations (ILS)*: 75 (6x6), 100 (10x10), 200 (20x20), 250 (30x30); *maximum number of iterations without improvement (ILS)*: 50; *number of repetitions (CH)*: 500 (6x6,10x10), and 1000 (20x20,30x30); *restricted candidate list size*: 5 (6x6,10x10), and 6 (20x20,30x30); *maximum neighbourhood size (LS and Pert)*: 6 (6x6), 10 (10x10), 20 (20x20) and 30 (30x30); *probability of removing a resource (Pert)*: 7.5%; *probability of adding a resource (Pert)*: 2.5%; *maximum number of modifications (Pert)*: $U(3, 5)$; *maximum number of failures (Pert)*: 100.

The MIP model (1)-(19) was solved by Gurobi 9.02 for instances 1 to 24 (6x6, 10x10 and 20x20) with five and six resources; instances with 900 nodes could not be solved due to out-of-memory errors. In all cases, the Gurobi time limit was set as 7200 seconds. The ILS metaheuristic was coded in Python 3.8 with the support of NetworkX library (NetworkX, 2021) for the network representation and manipulation; our code also relied on Dijkstra's shortest path algorithm from NetworkX (2021). The ILS was run five times for all instances. All results were obtained by a personal computer with an i7-9750H processor, 2.60GHz and 8.0GB of RAM.

5.2. Mixed integer programming

The results obtained by Gurobi when solving the MIP model (1)-(19) are shown in Tables 2 and 3, for five and six resources, respectively. These tables contain, for each instance, the objective function value (OFV), the lower bound (LB), the lower bound value at the root node (LB0), the total burned area, the number of nodes explored in the branch-and-bound tree, the number of integer so-

Table 1
Instances' parameters and information.

Ins- tance	Size	Wind Direction	Propagation Time [min] North South East West	Ignition Node	Burned Nodes	Candidates First Res.
1	6x6	Southeast	U(7,9) U(2,4) U(4,6) U(6,8)	(3,3)	11	6
2	6x6	Southeast	U(7,9) U(1,3) U(4,6) U(6,8)	(3,3)	11	9
3	6x6	Southeast	U(7,9) U(2,4) U(3,5) U(6,8)	(3,3)	9	10
4	6x6	Southeast	U(7,9) U(1,3) U(3,5) U(6,8)	(3,3)	11	12
5	6x6	South	U(7,9) U(2,4) U(4,6) U(4,6)	(3,3)	9	11
6	6x6	South	U(7,9) U(1,3) U(4,6) U(4,6)	(3,3)	12	11
7	6x6	South	U(7,9) U(2,4) U(3,5) U(3,5)	(3,3)	10	14
8	6x6	South	U(7,9) U(1,3) U(3,5) U(3,5)	(3,3)	12	16
9	10x10	Southeast	U(7,9) U(2,4) U(4,6) U(6,8)	(5,5)	10	8
10	10x10	Southeast	U(7,9) U(1,3) U(4,6) U(6,8)	(5,5)	11	12
11	10x10	Southeast	U(7,9) U(2,4) U(3,5) U(6,8)	(5,5)	11	9
12	10x10	Southeast	U(7,9) U(1,3) U(3,5) U(6,8)	(5,5)	14	12
13	10x10	South	U(7,9) U(2,4) U(4,6) U(4,6)	(5,5)	10	13
14	10x10	South	U(7,9) U(1,3) U(4,6) U(4,6)	(5,5)	12	17
15	10x10	South	U(7,9) U(2,4) U(3,5) U(3,5)	(5,5)	13	14
16	10x10	South	U(7,9) U(1,3) U(3,5) U(3,5)	(5,5)	17	18
17	20x20	Southeast	U(7,9) U(2,4) U(4,6) U(6,8)	(10,10)	8	8
18	20x20	Southeast	U(7,9) U(1,3) U(4,6) U(6,8)	(10,10)	10	13
19	20x20	Southeast	U(7,9) U(2,4) U(3,5) U(6,8)	(10,10)	8	12
20	20x20	Southeast	U(7,9) U(1,3) U(3,5) U(6,8)	(10,10)	11	16
21	20x20	South	U(7,9) U(2,4) U(4,6) U(4,6)	(10,10)	9	10
22	20x20	South	U(7,9) U(1,3) U(4,6) U(4,6)	(10,10)	12	18
23	20x20	South	U(7,9) U(2,4) U(3,5) U(3,5)	(10,10)	9	16
24	20x20	South	U(7,9) U(1,3) U(3,5) U(3,5)	(10,10)	14	22
25	30x30	Southeast	U(7,9) U(2,4) U(4,6) U(6,8)	(15,15)	9	13
26	30x30	Southeast	U(7,9) U(1,3) U(4,6) U(6,8)	(15,15)	13	15
27	30x30	Southeast	U(7,9) U(2,4) U(3,5) U(6,8)	(15,15)	11	12
28	30x30	Southeast	U(7,9) U(1,3) U(3,5) U(6,8)	(15,15)	14	19
29	30x30	South	U(7,9) U(2,4) U(4,6) U(4,6)	(15,15)	11	15
30	30x30	South	U(7,9) U(1,3) U(4,6) U(4,6)	(15,15)	16	18
31	30x30	South	U(7,9) U(2,4) U(3,5) U(3,5)	(15,15)	14	16
32	30x30	South	U(7,9) U(1,3) U(3,5) U(3,5)	(15,15)	19	24

Table 2
Results obtained for the MIP model solved with Gurobi - five resources.

Ins- tance	OFV	LB	LB0	Burned Nodes	Node Count	Solution Count	Time Best[s]	Total Time[s]
1	26	26	13.8	25	1	7	1.6	1.9
2	26	26	14.7	25	1	7	1.3	1.4
3	27	27	12.9	26	1	7	1.7	1.9
4	27	27	15.1	26	1	9	1.3	1.4
5	26	26	13.8	25	1	7	2.5	2.6
6	27	27	17.1	26	1	6	1.2	1.4
7	28	28	15.7	27	35	7	1.6	3.5
8	28	28	17.9	27	1	7	1.4	1.8
9	42	42	14.2	41	1	8	12.9	13.1
10	47	47	17.3	46	284	4	8.4	10.1
11	48	48	15.6	47	191	8	14.3	14.5
12	55	55	20.5	54	207	8	17.6	18.4
13	52	52	15.6	51	1251	9	20.2	20.4
14	56	56	19.7	55	1	6	13.5	13.8
15	59	59	19.4	58	221	7	13.8	14.2
16	64	64	25.3	63	220	7	16.0	16.4
17	–	37.5	12.1	–	441	0	–	93.6
18	–	48.7	15.6	–	7	0	–	64.6
19	–	41.2	12.7	–	13	0	–	62.7
20	–	55.3	17.1	–	13	0	–	77.5
21	–	46.3	13.8	–	7	0	–	96.5
22	–	63.3	19.1	–	3	0	–	91.3
23	–	15.2	15.2	–	0	0	–	11.8
24	–	21.9	21.9	–	0	0	–	11.1

lutions generated, the time for obtaining the best solution and the total runtime from Gurobi.

Problems with 36 nodes (instances 1 to 8) were promptly solved, not taking more than 3.8 seconds to achieve optimality. Problems with 100 nodes are still easy to be solved, not taking more than 30 seconds. In instances 20x20 with 5 resources (Table 2), an out-of-memory error halted the model execution before the imposed time limit in all cases. Particularly, in instances 23 and 24, the processing was interrupted at the root node. In instances 20x20 with 6 resources (Table 3), there were four cases that no integer solutions were generated at all, but only lower bounds. Concerning grids of size 30x30, no problem could be solved by Gurobi due to memory limitations from the very beginning of the model execution.

One may note that all cases where integer solutions are generated, the solutions are optimal. Regarding instances 17 to 24 with five resources, additional tests were made by running the mathematical model with the best ILS solutions as initial solutions. As a result, the optimal lower bound for instance 17 was achieved (LB = 40.0), and the lower bounds for instances 23 and 24 were signifi-

Table 3
Results obtained for the MIP model solved with Gurobi - six resources.

Ins- tance	OFV	LB	LB0	Burned Nodes	Node Count	Solution Count	Time Best[s]	Total Time[s]
1	23	23	13.0	22	1	7	1.7	1.8
2	24	24	14.2	23	1	6	1.9	2.0
3	24	24	12.3	23	16	6	2.6	3.5
4	24	24	14.4	23	1	5	1.4	1.6
5	24	24	13.1	23	12	7	2.7	3.8
6	25	25	16.4	24	1	6	1.3	1.6
7	24	24	15.0	23	1	7	2.0	2.0
8	25	25	17.2	24	1	7	2.0	2.1
9	38	38	13.0	37	440	4	17.6	17.9
10	43	43	15.4	42	1499	6	17.1	17.2
11	43	43	14.2	42	36	4	10.6	15.2
12	50	50	18.5	49	158	7	14.8	15.0
13	48	48	14.6	47	2250	10	25.0	25.2
14	52	52	18.6	51	1088	6	17.1	17.7
15	55	55	18.3	54	222	4	18.5	18.8
16	59	59	23.7	58	54	6	19.3	19.5
17	–	33	11.4	–	226,918	0	–	7,200.2
18	–	49	14.8	–	333,866	0	–	7,200.2
19	–	39	12.0	–	192,853	0	–	7,200.3
20	–	57	16.0	–	217,700	0	–	7,200.4
21	44	44	12.9	43	2400	1	273.8	274.4
22	63	63	18.1	62	6303	1	425.9	426.3
23	61	61	14.3	60	22,149	3	640.5	640.9
24	99	99	20.6	98	16,707	9	1,646.5	1,647.0

Table 4
ILS results for problems with five resources.

	Best solution (OFV)			Average values (5 replications)				Total (ILS)	
Instance	CH	LS	ILS	CH	LS	ILS	Time[s]	Best	Time[s]
1	27	27	26	27.0	26.8	26.0	1.2	5	6.1
2	27	26	26	27.0	26.2	26.0	1.2	5	5.9
3	28	27	27	28.2	27.2	27.0	1.0	5	5.2
4	28	27	27	28.0	27.0	27.0	1.0	5	5.0
5	27	27	26	27.0	27.0	26.0	1.1	5	5.7
6	27	27	27	27.0	27.0	27.0	1.0	5	4.8
7	28	28	28	28.0	28.0	28.0	1.0	5	4.8
8	28	28	28	28.0	28.0	28.0	1.0	5	5.1
9	44	42	42	44.6	42.0	42.	4.7	5	23.5
10	52	47	47	51.2	47.8	47.0	5.1	5	25.4
11	52	48	48	51.2	48.0	48.0	5.1	5	25.7
12	58	55	55	57.6	55.0	55.0	5.0	5	25.2
13	55	55	52	54.6	53.2	52.0	5.3	5	26.3
14	61	59	56	61.4	58.8	56.0	5.7	5	28.6
15	64	59	59	64.0	59.0	59.	4.7	5	23.7
16	69	65	64	69.0	65.0	64.2	5.8	4	29.1
17	42	40	40	42.0	40.0	40.0	32.5	5	162.7
18	61	59	56	61.4	59.4	56.0	39.5	5	197.7
19	52	51	49	52.0	50.4	49.0	37.5	5	187.4
20	80	80	73	80.6	77.2	73.0	34.7	5	173.5
21	59	51	51	58.8	51.0	51.0	33.8	5	168.9
22	83	75	75	79.4	75.0	75.0	32.8	5	163.8
23	75	75	73	75.8	73.4	73.0	35.7	5	178.6
24	117	116	113	119.0	116.0	112.8	41.2	3	206.1
25	48	48	46	47.6	46.8	46.0	82.1	5	410.4
26	72	65	64	70.4	64.2	64.0	75.6	5	378.1
27	57	57	56	58.0	57.0	56.0	83.3	5	416.7
28	94	88	79	94.2	88.0	79.0	79.4	5	396.8
29	66	66	61	64.4	62.0	61.0	72.8	5	364.2
30	90	88	84	90.0	85.6	84.0	78.8	5	394.2
31	90	90	86	89.0	88.6	86.2	86.9	4	434.5
32	143	132	124	143.4	133.6	127.2	98.8	3	494.2

cantly improved to 56.4 and 77.3, respectively. However, no significant lower bound improvements were observed for instances 18 to 22.

5.3. ILS Results

The ILS results are shown in Tables 4 and 5, for problems with five and six resources, respectively. In these tables, the first column indicates the instance number, followed by objective function value of the best solution produced by the constructive heuristic, by the local search procedure and by ILS. In the following three columns, the average objective function values over five runs are reported for the same three approaches. Afterwards, the average running time is reported, followed by the total number of times that the best solution was achieved (out of five), and the total running time, considering the five replications.

Table 5
ILS results for problems with six resources.

	Best solution (OFV)			Average values (5 replications)				Total (ILS)	
Instance	CH	LS	ILS	CH	LS	ILS	Time[s]	Best	Time[s]
1	25	25	23	24.6	24.4	23.6	1.2	2	6.2
2	26	25	24	25.4	24.4	24.0	1.2	5	5.9
3	25	24	24	24.8	24.0	24.0	1.2	5	6.1
4	27	24	24	26.8	24.4	24.0	1.7	5	8.6
5	25	24	24	24.4	24.2	24.0	1.2	5	5.9
6	25	25	25	25.0	25.0	25.0	1.1	5	5.7
7	26	25	24	26.0	25.0	24.0	1.5	5	7.7
8	27	25	25	26.2	25.0	25.0	1.2	5	6.0
9	41	38	38	40.4	38.0	38.0	5.1	5	25.4
10	47	45	43	46.6	43.8	43.0	5.3	5	26.4
11	45	43	43	44.8	43.0	43.0	4.7	5	23.7
12	56	53	50	55.8	51.8	50.4	6.0	3	29.8
13	50	49	48	50.2	49.0	48.0	5.2	5	26.0
14	57	53	52	56.4	53.6	52.6	6.0	2	30.0
15	58	55	55	57.4	55.0	55.0	5.7	5	28.6
16	66	62	59	67.0	61.4	59.0	7.8	5	39.2
17	36	35	33	36.6	34.6	33.0	39.9	5	199.5
18	58	54	49	57.4	52.2	49.8	38.5	3	192.6
19	49	47	39	47.8	43.8	39.8	39.3	4	196.4
20	74	67	57	74.8	67.4	57.0	39.8	5	198.9
21	49	44	44	49.6	44.4	44.0	36.7	5	183.4
22	74	64	63	74.8	64.4	63.0	41.1	5	205.4
23	68	61	61	67.8	61.8	61.0	37.	5	185.1
24	111	101	100	109.2	103.0	101.4	41,1	4	205.4
25	38	38	38	39.2	39.2	38.8	94.2	3	471.1
26	67	57	56	66.2	57.0	56.0	117.4	5	587.0
27	53	51	49	53.2	51.6	49.8	112.9	3	564.7
28	86	80	70	84.0	81.0	70.6	151.9	2	759.7
29	57	53	53	57.4	53.6	53.4	93.4	4	467.2
30	85	81	75	85.2	80.0	75.2	123.0	4	615.0
31	79	79	74	78.4	78.0	74.0	106.8	5	534.2
32	135	121	114	136.2	121.4	114.0	133.0	5	665.2

A comparison is made between the ILS objective function value and the lower bounds obtained from Gurobi for instances 1 to 24 (with five and six resources) to assess the ILS effectiveness. This set of 48 problems comprises 36 optimal solutions, five optimal lower bounds (instance 17 from Table 2 and instances 17 to 20 from Table 3), seven non-optimal lower bounds (instances 18 to 24 from Table 2). As previously noted, the lower bounds were improved in three cases after running the MIP model having as an initial solution the ILS solution. In summary, optimality was reached for 40 out of 41 cases, and the only non-optimal solution had a 1.0% gap. By considering the other seven cases (instances 18 to 24 with five resources), the overall average gap is 2.8%. Instances 30x30 cannot be compared to exact lower bounds.

A comparison between Gurobi and ILS regarding the processing time has to consider that Gurobi was processed aiming to achieve optimality. However, this was done by balancing between finding new feasible solutions and proving that the current solution is optimal. Moreover, a time limit of 7200 seconds was imposed for solving the MIP models, and, as a result, their processing times can be high. In fact, the average Gurobi runtime for instances 1–24 with five and six resources, excluding the out-of-memory cases, was 802.3 seconds. On the other hand, the ILS average processing time for the same instances, considering all replications, was 52.5 seconds.

In Tables 4 and 5 the objective function values are always integral, meaning that all resources were utilised. This was expected as the problem set proposed in this work was tested under a limited number of resources (five and six). If more resources were available, there could exist fractional optimal solutions. For example, if four resources were made available at instant 10 and four resources were made available at instant 15, the optimal solution for instance 1 would be 7.625, indicating seven burned nodes and five out of eight resources utilised ($5/8 = 0.625$). This example also considered that the ignition takes place at node (1,1).

Table 6
ILS performance by grid size.

Grid	ILS relative improvement over CH	ILS relative improvement over LS	Coef. Var. OFV	Coef. Var. Runtime	Average Runtime All Repl.[s]
6x6	3.5%	1.2%	0.1%	9.4%	5.9
10x10	6.7%	1.4%	0.1%	10.6%	27.3
20x20	10.2%	3.5%	0.6%	9.8%	187.8
30x30	8.6%	3.8%	0.8%	13.8%	497.1

An important figure is the overall average number of best solutions produced by ILS: 4.57 (out of five replications). This value varied according to the grid size in the following way: 4.81 (6x6), 4.63 (10x10, 20x20), 4.25 (30x30). Even for the cases where the best solution was not achieved, the variation is low. We further comment on the coefficient of variation of the objective function values below.

The summary of the average ILS performance is shown in Table 6. The first column indicates the grid size. The other columns display the average values of the relative improvement of the objective function values of ILS with respect to the constructive heuristic (i.e, $(z_{CH} - z_{ILS})/z_{CH}$) and to the local search (i.e., $(z_{LS} - z_{ILS})/z_{LS}$) approaches (columns 2 and 3, respectively), followed by the coefficient of variation of the objective function values and of the ILS runtime. One may note that the ILS relative improvement increases with the grid size. The coefficient of variation of the objective function values are very low, which is a significant achievement, while an admissible variation is observed in the ILS runtime.

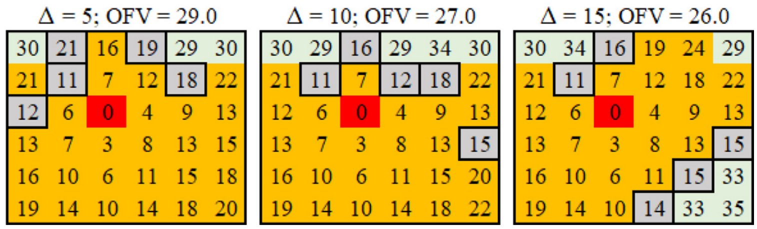

Fig. 3. Effect of different Δ values on instance 5.

5.4. Analysis of the fire suppression delay parameter

A foundational premise on which this research relies is that when suppression resources are deployed along a rasterised landscape, the nodes with resources become protected and hinder the fire propagation through them. The mathematical model addresses this issue by adding a delay to the fire propagation time from the protected node to its adjacent nodes. By properly adjusting the delay parameter (Δ) regarding the instant that the objective function is evaluated, one can guarantee that the suppression resources will act as fire blockers throughout the considered time horizon. In this case, the smallest value that the Δ parameter can assume is given by the difference between the instant the objective function is evaluated and the instant that resources are first released and plus one.

If Δ assumes values inferior to the above-indicated, the suppression resources will act as fire retardants rather than blockers. Such a situation occurs, for instance, when fire intensity is sufficiently high to prevent a fire crew from protecting the whole node area. Consequently, the fire path through the node may not be blocked but rather retarded. By adopting small Δ values (which can be node-specific), the modelling can thus address situations in which suppression resources are likely to fail to block an intense fire. Hence, optimisation approaches can indicate how to deploy suppression resources effectively.

In this regard, additional tests were made both with the mathematical model and the ILS metaheuristic. We considered the case with five resources and tested three different values for Δ (5, 10, 15), besides the value of 50, as in the previous section. Note that forcing Δ to be 20 (or larger) produces the same effect as 50, for reasons already given. To assess the ILS performance, we limited our experiments to landscapes 6x6, 10x10 and 20x20, which are solvable by Gurobi. As in the previous experiments, Gurobi was processed with a time limit of 7200 seconds, ILS was processed five times, and the minimum value for each instance was considered. The results are displayed in Table 7. In all cases, ILS reached the optimal solution. Therefore, we will concentrate our discussion on the effects of different Δ values rather than the heuristic performance (processing times and the number of best solutions). Note that in Table 7, the objective function values that appear in the last column come from the previous computational experiments, either from Table 2 (instances 1 to 16) or Table 4 (instances 17 to 24), as Gurobi failed to generate integer solutions with five resources for larger instances. All other values are optimal objective function values obtained both by Gurobi and ILS.

In Table 7, the average reduction in the number of burned nodes when Δ increases from five to ten is 10.3%, and the maximum reduction is 17.1%. Then, by increasing Δ from ten to fifteen, the average reduction in the number of burned nodes is 3.7%, and the maximum reduction is 11.1%. In both cases, the maximum reduction was observed for instance 18. The reduction in the average number of burned nodes is expected, as the larger the delay parameter, the more effective the protection provided by the suppression resources is.

Table 7
Objective function values for different Δ values and five resources.

Instance	OFV (Δ = 5)	OFV (Δ = 10)	OFV (Δ = 15)	OFV (Δ = 50)
1	29	26	26	26
2	29	26	26	26
3	30	27	27	27
4	30	27	27	27
5	29	27	26	26
6	29	27	27	27
7	31	28	28	28
8	31	28	28	28
9	49	44	42	42
10	55	49	47	47
11	57	51	49	48
12	61	56	55	55
13	58	54	52	52
14	63	58	56	56
15	68	62	60	59
16	71	66	64	64
17	51	43	40	40
18	77	64	57	56
19	61	52	49	49
20	90	82	74	73
21	65	55	51	51
22	93	84	77	75
23	86	76	73	73
24	120	115	113	113

Fig. 3 shows the influence of different Δ values on instance 5. For the three cases, the numbers indicate the fire arrival instants at each cell (node). The red cells are the ignition nodes; the orange cells are burned by the time the objective function is evaluated (instant 28). The grey cells are those where the resources were deployed, and the light green cells are protected. Interestingly, the retardant effect is observed for Δ = 5, for the cell (2;1) (second line, first column), with fire arrival instant 21. The fire first arrived at node (2;2) at instant 11. Considering that the fire spread time between nodes (2;2) and (2;1) equals to five (a value that only appears in the problem data set) and by adding the delay of five time units, the fire reaches node (2;1) at instant 21, despite suppression resources being placed in the surrounding nodes (2;2) and (3;1). For Δ = 5 and Δ = 10, the resources are mostly deployed in the upper parts of the landscape. As the fire propagates more rapidly in the southeast direction, it is advantageous to position resources in the upper nodes when Δ is small. However, for Δ = 15, one can observe that the lower part of the landscape becomes protected, and the retardant effect is sufficient for protection until the instant the objective function is evaluated. The objective function values 29, 27, and 26, indicate 28, 26, and 25 burned nodes, respectively. In all cases, all resources were utilised (which forces the second term of the objective function to be 1.0).

6. Conclusions

Combating wildfires is a major concern throughout the world. With the ever-increasing occurrence and intensity of forest fires, it is paramount to extinguish them as quickly as possible before they become uncontrollable. A particular concern relates to forest fire occurrences in wildlands surrounding areas with human settlements, posing a greater risk to human lives.

In an attempt to effectively respond to fire occurrences, fire management teams have mapped fire-prone areas concerning their spatial data, raster data from geographic information systems, fuels and typical weather conditions (Landfire, 2022). These data allow determining the fire spread rates between adjacent nodes of a rasterised landscape, which in turn allows applying optimisation-based fire suppression planning tools.

In this regard, we addressed a fire suppression problem aiming to determine where to position the suppression resources that become available in different time instants, with the objective of the minimising the number of burned nodes by a target instant and the total number of resources as a secondary objective.

We proposed an iterated local search metaheuristic that can solve large instances of the problem in short computing times. The approach was validated by comparing the objective function values with those from a mixed-integer programming model from the literature. ILS found provably optimal solutions for grid sizes ranging from 36 to 400 nodes, with very small coefficients of variation, attesting that the method is robust. Problems with 900 nodes were also solved with reasonable computing times. An important feature of the proposed approach is the optimisation of resource positioning, taking into account its influence on the spatial and temporal propagation of fire. This collaborative approach allows assessing the effect of positioning suppression resources in the fire spread behaviour, thus supporting fire management decisions during the initial attack.

The proposed approach can be directly extended to address variants such as the existence of different resource types or the objective of asset protection. A relevant extension is to model the movement of resources (taking into account that fire blocks paths), or even test a fire spread simulator other than MTT solved with Dijkstra's algorithm. The inclusion of uncertainty in the fire spread simulator is a natural follow-up to this work.

Acknowledgements

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319/2020 and within project PCIF/GRF/0141/2019 “O3F - An Optimization Framework to Reduce Forest Fire”. This paper has greatly benefitted from the insights and suggestions of anonymous reviewers on an earlier version of the paper.

References

Alvelos, F. (2018). Mixed integer programming models for fire fighting. In *International conference on computational science and its applications* (pp. 637–652). Melbourne, Australia: Springer.

Belval, E. J., Wei, Y., & Bevers, M. (2015). A mixed integer program to model spatial wildfire behavior and suppression placement decisions. *Canadian Journal of Forest Research, 45*(4), 384–393.

Belval, E. J., Wei, Y., & Bevers, M. (2019). Modeling ground firefighting resource activities to manage risk given uncertain weather. *Forests, 10*(12), 1077.

Blum, C., Blesa, M. J., García-Martínez, C., Rodríguez, F. J., & Lozano, M. (2014). The firefighter problem: application of hybrid ant colony optimization algorithms. In *European conference on evolutionary computation in combinatorial optimization* (pp. 218–229). Granada, Spain: Springer.

Chi, S.-D., Lim, Y.-H., Lee, J.-K., Lee, J.-S., Hwang, S.-C., & Song, B.-H. (2003). A simulation-based decision support system for forest fire fighting. In *Congress of the italian association for artificial intelligence* (pp. 487–498). Pisa, Italy: Springer.

Chung, W. (2015). Optimizing fuel treatments to reduce wildland fire risk. *Current Forestry Reports, 1*(1), 44–51.

Develin, M., & Hartke, S. G. (2007). Fire containment in grids of dimension three and higher. *Discrete Applied Mathematics, 155*(17), 2257–2268.

Dimopoulou, M., & Giannikos, I. (2001). Spatial optimization of resources deployment for forest-fire management. *International Transactions in Operational Research, 8*(5), 523–534.

Donovan, G. H., & Rideout, D. B. (2003). An integer programming model to optimize resource allocation for wildfire containment. *Forest Science, 49*(2), 331–335.

Duff, T. J., & Tolhurst, K. G. (2015). Operational wildfire suppression modelling: A review evaluating development, state of the art and future directions. *International Journal of Wildland Fire, 24*(6), 735–748.

Dunn, C. J., Thompson, M. P., & Calkin, D. E. (2017). A framework for developing safe and effective large-fire response in a new fire management paradigm. *Forest Ecology and Management, 404*, 184–196.

Fernandes, P. M., Pacheco, A. P., Almeida, R., & Claro, J. (2016). The role of fire-suppression force in limiting the spread of extremely large forest fires in portugal. *European Journal of Forest Research, 135*(2), 253–262.

Figueira, G., & Almada-Lobo, B. (2014). Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory, 46*, 118–134.

Finbow, S., & MacGillivray, G. (2009). The firefighter problem: A survey of results, directions and questions. *Australasian Journal of Combinatorics, 43*, 57–77.

Finney, M. A. (1998). FARSITE, Fire Area Simulator–model development and evaluation. *Technical Report*. United States Department of Agriculture.

Finney, M. A. (2002). Fire growth using minimum travel time methods. *Canadian Journal of Forest Research, 32*(8), 1420–1424.

Gendreau, M., Potvin, J.-Y., et al., (2010). *Handbook of metaheuristics vol. 2*. Springer.

Haight, R. G., & Fried, J. S. (2007). Deploying wildland fire suppression resources with a scenario-based standard response model. *INFOR: Information Systems and Operational Research, 45*(1), 31–39.

Hof, J., Omi, P. N., Bevers, M., & Laven, R. D. (2000). A timing-oriented approach to spatial allocation of fire management effort. *Forest Science, 46*(3), 442–451.

HomChaudhuri, B., Kumar, M., & Cohen, K. (2013). Genetic algorithm based simulation–optimization for fighting wildfires. *International Journal of Computational Methods, 10*(06), 1350035.

Hu, B., Windbichler, A., & Raidl, G. R. (2015). A new solution representation for the firefighter problem. In *European conference on evolutionary computation in combinatorial optimization* (pp. 25–35). Copenhagen, Denmark: Springer.

Jewell, W. S. (1963). Forest fire problems—a progress report. *Operations Research, 11*(5), 678–692.

Kirsch, A. G., & Rideout, D. B. (2005). Optimizing initial attack effectiveness by using performance measures. In *Proceedings of the 2003 symposium: Systems analysis in forest resources* (pp. 183–188). United States Department of Agriculture.

Landfire (2022). Landscape Fire and Resource Management Planning Tools of the U.S. Department of Agriculture Forest Service and U.S. Department of the Interior. https://www.landfire.gov/, Accessed: 2022-04-06.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In *Handbook of metaheuristics* (pp. 320–353). Springer.

MacLellan, J. I., & Martell, D. L. (1996). Basing airtankers for forest fire control in Ontario. *Operations Research, 44*(5), 677–686.

Marianov, V., & ReVelle, C. (1992). The capacitated standard response fire protection siting problem: Deterministic and probabilistic models. *Annals of Operations Research, 40*(1), 303–322.

Martell, D. L. (1982). A review of operational research studies in forest fire management. *Canadian Journal of Forest Research, 12*(2), 119–140.

Martell, D. L. (2015). A review of recent forest and wildland fire management decision support systems research. *Current Forestry Reports, 1*(2), 128–137.

Martí, R., Resende, M. G. C., & Ribeiro, C. C. (2013). Multi-start methods for combinatorial optimization. *European Journal of Operational Research, 226*(1), 1–8.

Mees, R., & Strauss, D. (1992). Allocating resources to large wildland fires: A model with stochastic production rates. *Forest Science, 38*(4), 842–853.

Mees, R., Strauss, D., & Chase, R. (1993). Modeling wildland fire containment with uncertain flame length and fireline width. *International Journal of Wildland Fire, 3*(3), 179–185.

van der Merwe, M., Minas, J. P., Ozlen, M., & Hearne, J. W. (2015). A mixed integer programming approach for asset protection during escaped wildfires. *Canadian Journal of Forest Research, 45*(4), 444–451.

Michalak, K. (2014). Auto-adaptation of genetic operators for multi-objective optimization in the firefighter problem. In *International conference on intelligent data engineering and automated learning* (pp. 484–491). Salamanca, Spain: Springer.

Michalak, K. (2017). Ed-ls–a heuristic local search for the multiobjective firefighter problem. *Applied Soft Computing, 59*, 389–404.

Minas, J. P., Hearne, J. W., & Handmer, J. W. (2012). A review of operations research methods applicable to wildfire management. *International Journal of Wildland Fire, 21*(3), 189–196.

Minas, J. P., Hearne, J. W., & Martell, D. L. (2014). A spatial optimisation model for multi-period landscape level fuel management to mitigate wildfire impacts. *European Journal of Operational Research, 232*(2), 412–422.

NetworkX (2021). Network analysis in python. https://networkx.org/, Accessed: 2021-01-30.

Pappis, C. P., & Rachaniotis, N. P. (2010). Scheduling a single fire fighting resource with deteriorating fire suppression times and set-up times. *Operational Research, 10*(1), 27–42.

Rachaniotis, N. P., & Pappis, C. P. (2006). Scheduling fire-fighting tasks using the concept of "deteriorating jobs". *Canadian Journal of Forest Research, 36*(3), 652–658.

Rachaniotis, N. P., & Pappis, C. P. (2011). Minimizing the total weighted tardiness in wildfire suppression. *Operational Research, 11*(1), 113–120.

Ramos, N., de Souza, C. C., & de Rezende, P. J. (2020). A matheuristic for the firefighter problem on graphs. *International Transactions in Operational Research, 27*(2), 739–766.

Rideout, D., Wei, Y., & Kirsch, A. (2011). Optimal allocation of initial attack resources to multiple wildfire events. *International Journal of Safety and Security Engineering, 1*(3), 312–325.

Rodríguez-Veiga, J., Ginzo-Villamayor, M. J., & Casas-Méndez, B. (2018). An integer linear programming model to select and temporally allocate resources for fighting forest fires. *Forests, 9*(10), 583.

Roozbeh, I., Ozlen, M., & Hearne, J. W. (2018). An adaptive large neighbourhood search for asset protection during escaped wildfires. *Computers & Operations Research, 97*, 125–134.

Shephard, R. W., & Jewell, W. S. (1961). Operations research in forest fire problems. In A study of fire problems (pp. 145–165). National Academy of Science-National Research Council, Committee on Fire. https://www.nap.edu/initiative/committee-on-fire-research-and-the-fire-research-conference-of-the-national-academy-of-sciences.

Thompson, M. P., Silva, F. R., Calkin, D. E., & Hand, M. S. (2017). A review of challenges to determining and demonstrating efficiency of large fire management. *International Journal of Wildland Fire, 26*, 562–573.

UNDRR (2019). 2018 extreme weather events affected 60 million people. https://www.undrr.org/news/2018extreme-weather-events-affected-60m-people, Accessed: 2019-01-24.

Wei, Y., Rideout, D. B., & Hall, T. B. (2011). Toward efficient management of large fires: A mixed integer programming model and two iterative approaches. *Forest Science, 57*(5), 435–447.

Wei, Y., Thompson, M. P., Belval, E., Gannon, B., Calkin, D. E., & O'Connor, C. D. (2021). Comparing contingency fire containment strategies using simulated random scenarios. *Natural Resource Modeling, 34*(1), e12295.

Wei, Y., Thompson, M. P., Haas, J. R., Dillon, G. K., & O'Connor, C. D. (2018). Spatial optimization of operationally relevant large fire confine and point protection strategies: Model development and test cases. *Canadian Journal of Forest Research, 48*(5), 480–493.

Wei, Y., Thompson, M. P., Scott, J. H., O'Connor, C. D., & Dunn, C. J. (2019). Designing operationally relevant daily large fire containment strategies using risk assessment results. *Forests, 10*(4), 311.

Wiitala, M. R. (1999). A dynamic programming approach to determining optimal forest wildfire initial attack responses. In A. González-Cabán, & P. N. Omi (Eds.), *Symposium on fire economics, planning, and policy: Bottom lines* (pp. 115–123). San Diego, United States of America: USDA Forest Service, Pacific Southwest Research Station.

Wu, P., Cheng, J., & Feng, C. (2019). Resource-constrained emergency scheduling for forest fires with priority areas: An efficient integer-programming approach. *IEEJ Transactions on Electrical and Electronic Engineering, 14*(2), 261–270.

Zambon, M. J., de Rezende, P. J., & de Souza, C. C. (2018). Finding exact solutions for the geometric firefighter problem in practice. *Computers & Operations Research, 97*, 72–83.