

A Mathematical Programming Approach for a Wildfire Suppression Problem

Bibiana Granda-Chico^{1*}, Begoña Vitoriano¹ and José Rui Figueira²

^{1*}HUMLOG Research Group, Interdisciplinary Mathematics Institute (IMI), Plaza de Ciencias, 3, Madrid, 28040, Spain.

²CEGIST, Instituto Superior Técnico, Universidade de Lisboa, Street, Lisbon, 1049 - 001, Portugal.

*Corresponding author(s). E-mail(s): bibianag@ucm.es;
Contributing authors: bvitoriano@ucm.es; figueira@tecnico.ulisboa.pt;

Abstract

Wildfires are natural recurrent events, that may be devastating if not addressed correctly. In these situations, where quick and accurate decisions are needed, Operations Research can be helpful for providing fast and robust solutions. This paper focuses on the response actions taken during the suppression stage of a wildfire. A mixed integer linear programming model is proposed to obtain a wildfire suppression strategy, including the wildfire behaviour changes induced by the solution. The selected wildfire suppression strategy is modelled in detail, pointing out which locations to control and their timing, based on available paths between them, avoiding engagement in dangerous situations. A computational study is carried out to determine the most suitable solver to provide exact solutions of the model. Also, a two-stage version of the model is proposed to deal with the multicriteria nature of the problem. A case study is also included to validate the model's applicability, which is solved using the two proposed versions of the model and an iterative approach to compare their performance.

Keywords: Wildfire management, Wildfire suppression, Optimization, Mixed Integer Linear Programming

1 Introduction

Wildfires are natural phenomena leading to natural disasters. As some recent studies show, they are increasing in severity and frequency over the last decades. The reasons for this include large accumulations of fuel on the landscape (Curt & Fréjaville, 2017) or the global warming, which fosters extreme weather situations in which larger fires occur (Shi et al., 2021). This leads to an increasingly flammable landscape since forests are more prone to crown fire during their period of regeneration (Zylstra, 2018).

In this regard, wildfire suppression plays a key role in forests' care and community and assets protection. Wildfire suppression is a compendium of tasks related to the direct interaction with wildfires, determining where and when to act in order to stop the spread of wildfires.

On account of the increasing wildfire activity and severity, wildfire suppression costs have risen (Ingalsbee & Raja, 2015), and they will continue to escalate due to the growth of house density, which leads to a larger wildland-urban interface, in which a higher number of houses may be threatened (Bayham & Yoder, 2020). This rise in costs may be dampened by improving the efficiency of the existing resources (Belval, Stonesifer, & Calkin, 2020) or by designing efficient strategies for wildfire suppression considering weather, fuel and topography for creating containment opportunities (Fernandes, Pacheco, Almeida, & Claro, 2016).

Operations Research (OR) has already proven to be a good tool for decision support in a number of disciplines. Wildfire suppression can also benefit from these methodologies to improve the efficiency of the mentioned tactics. Granda, León, Vitoriano, and Hearne (2023) provide a good insight of how OR models in literature deal with decisions related to wildfire suppression.

The aim of this paper is to develop a mathematical programming model that supports wildfire suppression decisions, optimizing limited ground resources (a brigade) during the dispatch stage of the wildfire suppression. The model has been tested for different landscape sizes and using three different solvers to determine their performance in a computational study with randomly generated instances. Finally, a case study is presented based on a simulated wildfire with FLAMMAP (Finney et al., 2006) on a real landscape.

The structure of the paper is as follows: Section 2 presents a literature review of the most meaningful papers using OR applied to wildfire suppression. Section 3 describes the methods used to develop the model presented in this paper. Section 4 summarizes the mathematical formulation of the model. In Section 5, the results of the computational study can be found. Section 6 describes the multicriteria methodology used to deal with the different objectives of the model, including an analysis of the results obtained with the alternative approaches. A case study is included in section 7 to complete the model's validation. In Section 8 conclusions are drawn.

2 Wildfire suppression in OR literature

Wildfire suppression is a term that gathers all the actions devoted to extinguish a wildfire. Decisions on those actions should consider natural factors, such as weather conditions or terrain features, which determine the wildfire behaviour. However, it

also will be affected by previous human decisions. Wildfire suppression management is not only about dealing with the wildfire directly, but also planning, acquiring and arranging resources that will be necessary when the wildfire starts. As disasters, wildfires can be dealt with using the stages of the disaster management cycle (Tomasini & van Wassenhove, 2009). The wildfire suppression term comprehends actions taken in the response phase, but also in the prevention and preparedness phase of the disaster.

Models dealing with resource preparedness can be found in Wei, Belval, Thompson, Calkin, and Stonesifer (2017), Chow and Regan (2011) or Suarez, Gomez, Medaglia, and Akhavan-Tabatabaei (2016). Such models make plans prior to the occurrence of a wildfire. They usually rely on risk maps developed basing on indices of wildfire hazard (Ríos-Mercado, González-Tagle, Díaz-Romero, & Rodríguez-y Silva, 2020; Zeferino, 2020).

Regarding response, a more dynamic approach is needed. Instead of using static parameters such as hazard indexes, some authors make use of data related to actual wildfire behaviour such as maximum spread direction and velocity. These data – often provided by simulators – are used to determine the final shape of the wildfire and optimize the available resources to minimize its impact (Donovan & Rideout, 2003; Hu & Ntamo, 2009; Rodríguez-Veiga, Guinzo Villamayor, & Casas Mendez, 2018; Yang, Guo, & Yang, 2019). However, this methodology leads to an overestimation of the resources needed. The final shape of the wildfire is calculated given a free burn scenario whereas suppression is applied, so the actual final shape will be smaller and thus less resources needed. In this regard, several authors use the Minimum Travel Time (MTT) methodology to integrate the wildfire behaviour into the optimization model, leading to a more realistic approach in which the wildfire suppression strategy affects the wildfire behaviour, hindering its expansion (Belval & Wei, 2019; Belval, Wei, & Bevers, 2015; Mendes & Alvelos, 2023; Wei, Rideout, & Hall, 2011; Wei et al., 2021). Nevertheless, some of them only determine where to locate the controls to stop the spreading of the wildfire, but do not deal with the exact timing of the wildfire suppression strategy (Belval et al., 2015; Mendes & Alvelos, 2023; Wei et al., 2011). Wei et al. (2011) and Mendes and Alvelos (2023) divide the time-span in intervals and consider that resources are made available as time goes by. Resources not placed in a certain interval are still available to be used in subsequent ones. However, only the total number of controls within each interval is controlled, not ensuring the avoidance of controls' simultaneity. Besides, the travel time of the brigades or the time it takes to construct a fireline are not negligible and should be accounted for safety reasons. Hence, some other authors consider also completion and travel times, providing a more realistic strategy that includes the timing of the suppression activities (Wei et al., 2021) and even the actual movement of the crews (Belval & Wei, 2019; Homchaudhuri, Kumar, & Cohen, 2013). This timing, along with the determination of the wildfire intensity, allows to set safety constraints, forbidding the engagement of the crews in sites which will be reached promptly by the wildfire (Belval & Wei, 2019; Wei et al., 2021; Wei, Thompson, Scott, O'Connor, & Dunn, 2019) or where the flame length is above a threshold that is considered dangerous (Wei et al., 2019). Wei et al. (2019) include a second optimization stage, in which the narrowest temporal gap between fire line completion and fire arrival time is maximized. However, they work

with precalculated fire arrival times, not considering the interaction between wildfire behaviour and the suppression strategy.

The wildfire suppression problem can be related to the vehicle routing problem with time windows (VRPTW) (Solomon, 1987). In fact, some authors propose solving a VRP to visit prioritized locations for wildfire suppression (Bodaghi, Palaneeswaran, Shahparvari, & Mohammadi, 2020; Shahidi, Ramezani, & Shahparvari, 2022; Yang et al., 2019). To the best of our knowledge, Belval and Wei (2019) is the first wildfire suppression paper that considers a variable safety buffer between the brigades and the fire. This safety buffer is essentially a time window that must be observed at each node, which depends on fire arrival times. However, the authors do not relate it to the VRP. The VRP has been extensively studied in the literature, and many variants with different features have arisen (Toth & Vigo, 2002).

In this paper a model is presented that, following the approach by Belval and Wei (2019), integrates wildfire spread and wildfire suppression. The MTT methodology is used to calculate dynamic fire arrival times which influence, but also are modified by the suppression strategy. The model describes in detail the brigade’s path, including the timing of the controls, considering both travel and controlling times, that determine the best suppression strategy. Belval and Wei (2019) present a case study to show the performance of the model, however, they conclude that commercial solvers such as CPLEX are not capable of solving it to optimality in reasonable running times. In this piece of research, a complete computational study is carried out, using different landscape features and sizes to fully investigate the model’s capabilities. The novelty of this paper is the proposal of a two-stage approach to deal with the multi-criteria nature of the problem, along with improvements in the formulation to reduce solving times. Besides, an iterative approach is explored to deal with large instances.

3 Model features and methodology

The model presented in this paper tackles the wildfire suppression problem in a dynamic way, considering the interaction between the fire spread and the suppression strategy. The goal is to determine the best ground suppression strategy, which comprises the movement of the brigade across the landscape and the actions to be carried out. Safety constraints are satisfied by providing detailed timing of these suppression actions.

In order to model the landscape, the space is divided into parcels. An assumption is that the wildfire will not escape the perimeter determined by the grid. This assumption is based on the fact that usually there are natural barriers to the spread of the fire. Besides, wildfires often have a maximum spread direction, not advancing – or advancing very slowly – towards the opposite direction. The ability of the wildfire to reach one parcel from another is determined by an adjacency matrix. Some parcels might be non-flammable, that is they cannot burn. The suppression strategy may take advantage of this kind of parcels, which in reality represent roads, lakes, etc., assuming that the wildfire cannot jump across them. This is a simplification, since non-linear propagation is frequent in real wildfires due to flying ashes. However, this model works only with ground wildfires, where this kind of behaviour is less usual.

Given the landscape divided into parcels, the wildfire suppression problem involves minimizing the value of the burned parcels with the available resources – in this case one brigade – once a wildfire has started. This brigade shall move along a continuous path, crossing parcels that may be controlled or not. When a *control* is placed in a parcel it is assumed to delay the spread of fire. In this study, the delay is assumed to be big enough to stop the fire spread completely through that parcel. This can be seen as a reasonable assumption since the model is prepared to work with ground wildfires, which are usually easier to stop placing physical barriers.

There are two main groups of constraints, one deals with the wildfire expansion and the other with the suppression strategy. However, they cannot be separated completely, because of the interrelation between the two processes.

To include the behaviour of the wildfire within the model, the time it takes the fire to travel from one parcel to the adjacent ones is needed as an input. This information can be calculated from data provided by a simulator, or other means. In this case it is calculated using the methodology from [Wei et al. \(2011\)](#) and [Belval et al. \(2015\)](#). The necessary data is taken from FLAMMAP ([Finney et al., 2006](#)). Then, the fire arrival times are modelled based on the MTT methodology, which assumes that the wildfire moves through the shortest paths ([Wei et al., 2011](#)). These shortest paths can be modified by the suppression strategy. The fire intensity also depends on the binding paths of the fire, so it may also change due to the suppression strategy ([Belval et al., 2015](#)). The necessary parameters to calculate the fire intensity in each parcel are also estimated, for instance, using a simulator. In the proposed model, the fire intensity is just calculated as a consequence of the fire arrival and does not influence the decisions to be made.

Regarding the suppression strategy, the decision to be made is which parcels must be controlled and the path that the brigade should follow to place the necessary controls. A parcel might be visited by the brigade but not controlled, only to access other parcels. The timing of the controls and visiting parcels must be monitored for safety reasons. A brigade cannot access a parcel after it has started to burn, and a parcel can be controlled only if the controlling actions are finished before the fire arrives. Given the structure of the model, only one visiting instant can be tracked. So only one visit is allowed to each parcel, the movement being governed by flow constraints.

Thus, the variables associated to the wildfire behaviour are the time instant when the wildfire reaches a parcel, the binding paths of the wildfire, the intensity of the wildfire in each parcel and if the parcel is burned. All those variables are subject to the wildfire suppression strategy, which variables determine the time and location of controls, monitoring the path the brigade follows. The initial and final location of the brigade's path are also variables.

The criteria to determine the best solution are the value of the total area burned and the associated costs of controlling a parcel, which will be minimized, along with the total time of operation. This characterizes the problem as multicriteria. The priority is to minimize the value of the area burned, then the costs of placing the controls and finally the total operational time. These criteria could be combined and studied to

determine a Pareto front, however, to carry out the computational study, a scalarization has been applied for comparing the instances. The three objectives are combined into a single objective function giving weights to the criteria. This scalarization could also be useful for operational application, since decision on wildfire suppression must be quick, which means that studying the Pareto front for each case could not be viable. The necessary weights could be obtained in advance in an interactive way with practitioners from historical and/or simulated cases.

A second test has been performed using a two-stage approach, in which the criteria are divided in two objective functions: the first stage minimizes the area burned and the second minimizes the controlling costs and operational time. In this way priority is given to minimize the area burned, whereas controlling costs and operational time are minimized given an upper bound on the previous objective.

4 A mixed integer linear programming model for wildfire suppression

In this section, the mathematical formulation of the model is presented. First, the basic model is introduced, and then, the last subsection is devoted to including some additional constraints that are found helpful to improve the model runtimes.

4.1 Indices and sets

In this model indices are basically used to refer to the parcels and some sets are used to group several parcels regarding its characteristics: to determine the ignition parcels or to group the adjacent parcels.

- $i, j = 1, \dots, n$: indices used for the parcels;
- A_i : set of all adjacent parcels to i , $i = 1, \dots, n$;
- P : set of all ignition parcels.

4.2 Parameters

Most of the parameters are related to the wildfire behaviour and to the characteristics of the brigade and their actions.

- a_{ij}^f : time it takes the fire to travel from i to j , $i = 1, \dots, n$, $j \in A_i$;
- a_{ij}^b : time it takes the brigade to travel from i to j , $i = 1, \dots, n$, $j \in A_i$;
- a_i^c : time it takes to control the parcel i , $i = 1, \dots, n$;
- a_i^{min} : time it takes to arrive to parcel i if it is the first one to be visited, $i = 1, \dots, n$;
- b_{ij} : intensity the fire reaches¹ in j if the fire comes from i , $i = 1, \dots, n$, $j \in A_i$;
- c_i : intensity of fire in i if wildfire ignites in that parcel, $i = 1, \dots, n$;
- M, N type parameters: positive big enough numbers;
- d_i^f : lost value if parcel i burns, $i = 1, \dots, n$;

¹Due to topographic features and wind direction the intensity of the fire varies depending on the arrival direction

- d_i^b : cost of controlling parcel i , $i = 1, \dots, n$;
- d^t : cost of unit time of brigade movement.
- w^f : weight of burned parcels;
- w^b : weight of controlled parcels;
- w^t : weight of brigade movement time.

4.3 Decision variables

The group of decision variables in the model comprises continuous variables for times and binary for decision making and states. Variables representing fire intensity are also continuous.

- x_i^f : time instant when the fire reaches the parcel i , $i = 1, \dots, n$;
- x_i^b : brigade arrival time to parcel i , $i = 1, \dots, n$;
- v_i : the intensity of fire when the fire reaches parcel i , $i = 1, \dots, n$;
- y_i : 1 if parcel i burns, 0 otherwise, $i = 1, \dots, n$;
- z_i : 1 if parcel i is controlled, 0 otherwise, $i = 1, \dots, n$;
- w_{ij} : 1 if fire spreads from parcel i to parcel j , 0 otherwise, $i = 1, \dots, n$, $j \in A_i$;
- r_{ij} : 1 if brigade travels from parcel i to parcel j , 0 otherwise, $i = 1, \dots, n$, $j \in A_i$;
- s_i : 1 if parcel i is the first parcel visited by the brigade, 0 otherwise, $i = 1, \dots, n$;
- t_i : 1 if parcel i is the last parcel visited by the brigade, 0 otherwise, $i = 1, \dots, n$.

4.4 Constraints

Constraints in the model can be classified in two groups. The first group of constraints describe how the wildfire spreads across the landscape, and the second group describes the wildfire suppression strategy defined by the controls' location, and the movements of the brigade, regarding the availability of the parcels to be crossed or controlled. This availability depends on the fire arrival times.

4.4.1 Wildfire spread constraints

These constraints belong to the first group and model the wildfire behaviour considering also the variables related to the wildfire suppression strategy.

- *Wildfire starting constraints.* Fire arrival time for ignition parcels is set to 0:

$$x_i^f = 0, \quad i \in P \quad (1)$$

- *Wildfire arrival time constraints.* These constraints model the wildfire arrival time. When a parcel i is not controlled (i.e., when $z_i = 0$), the fire arrival time will be the minimum from the adjacent parcels. If the parcel is controlled

(i.e., when, $z_i = 1$) the fire does not reach it and the constraint is not binding:

$$x_i^f \leq x_j^f + a_{ji}^f + Mz_i, \quad i = 1, \dots, n, \quad j \in A_i \quad (2)$$

$$x_i^f \geq x_j^f + a_{ji}^f w_{ji} - M(1 - w_{ji}), \quad i = 1, \dots, n, \quad j \in A_i \quad (3)$$

These two sets of constraints determine accurately the wildfire arriving times and the binding path of the fire.

- *Wildfire arrival constraints.* These constraints state that if the fire reaches a parcel, it will burn, and the arrival will be from one of the adjacent parcels:

$$\sum_{j \in A_i} w_{ji} = y_i, \quad i = 1, \dots, n, \quad i \notin P \quad (4)$$

- *Burning time constraint.* These constraints state that if a parcel is not burned, fire arrival time is set to be over a threshold, out of the range of fire:

$$x_i^f \geq M(1 - y_i), \quad i = 1, \dots, n \quad (5)$$

- *Wildfire intensity constraints.* These constraints determine the intensity of the fire in each parcel reached by the fire, which depends on the fire path:

$$v_i = \sum_{j \in A_i} w_{ji} b_{ji}, \quad i = 1, \dots, n, \quad i \notin P \quad (6)$$

- *Wildfire intensity constraints for ignition parcels.* These constraints determine the intensity of the wildfire in each ignition parcel, which only depends on the characteristics of that parcel:

$$v_i = c_i, \quad i \in P \quad (7)$$

4.4.2 Wildfire suppression constraints

The following constraints belong to the second group, modelling the wildfire suppression strategy (parcels to be controlled) and the brigade's movement.

- *Flow constraints.* These constraints model the fact that the brigade enters and exits the parcel the same number of times:

$$\sum_{j \in A_i} r_{ji} + s_i = \sum_{j \in A_i} r_{ij} + t_i, \quad i = 1, \dots, n \quad (8)$$

- *Non-overlapping brigade path constraints.* These constraints limit the number of times that a parcel can be visited by the brigade. The first bunch of them limits to one the number of times the brigade arrives to the parcel and the

second the number of times the brigade exits the parcel:

$$\sum_{j \in A_i} r_{ji} + s_i \leq 1, \quad i = 1, \dots, n \quad (9)$$

$$\sum_{j \in A_i} r_{ij} + t_i \leq 1, \quad i = 1, \dots, n \quad (10)$$

- *Brigade initial location constraints.* This constraint forces to be one and exactly one starting parcel for the wildfire suppression:

$$\sum_{i=1}^n s_i = 1 \quad (11)$$

- *Brigade final location constraints.* This constraint forces to be one and exactly one final parcel for the wildfire suppression:

$$\sum_{i=1}^n t_i = 1 \quad (12)$$

- *Controls constraints.* These constraints force that controls can only be placed in parcels that are visited by the brigade:

$$z_i \leq \sum_{j \in A_i} r_{ji} + s_i, \quad i = 1, \dots, n \quad (13)$$

- *Brigade binding path constraints.* These constraints are similar to the *Wildfire spread binding path constraints* (3). They give value to the variable that defines the path the brigade follows during the suppression strategy. If the brigade travels from j to i the time the brigade arrives to i is greater than the time the brigade arrives to j plus the travel time from j to i (also accounts for the time it takes to control j in case it is controlled):

$$x_i^b \geq x_j^b + a_j^c z_j + a_{ji}^b - N(1 - r_{ji}), \quad i = 1, \dots, n, \quad j \in A_i \quad (14)$$

- *Safety constraints.* These constraints avoid the brigade getting too close to the fire, considering the time controlling a parcel and the time it takes the brigade leaving. For these reasons, the control must be finished before the wildfire reaches the adjacent parcels:

$$x_i^b + a_i^c z_i \leq x_j^f, \quad i = 1, \dots, n, \quad j \in A_i \quad (15)$$

- *Non-visited parcel time constraints.* These constraints set that if the parcel is not visited, the visit-time is 0, otherwise there is a sufficiently large bound

(N''):

$$x_i^b \leq N \left(\sum_{j \in A_i} r_{ji} + s_i \right), \quad i = 1, \dots, n \quad (16)$$

- *Minimum arrival time constraints.* These constraints model the minimum time to reach the first parcel to start the wildfire suppression strategy:

$$x_i^b \geq a_i^{\min} s_i, \quad i = 1, \dots, n \quad (17)$$

This constraint can be modified to tighten the formulation:

$$x_i^b \geq a_i^{\min} \left(s_i + \sum_{j \in A_i} r_{ji} \right), \quad i = 1, \dots, n \quad (18)$$

This modification guides the algorithm in such a way that if the parcel i is visited, the time of visit should be at least $a_i^{\min}$. Equation (17) only sets a lower bound for the time of visit of the first visited parcel, whereas (18) also adds a lower bound for subsequently visited parcels.

4.5 Improved formulation

The objective of the problem is to find where to place the controls, considering that the brigade should travel from one parcel to another. Thus, the suppression strategy is assumed to start in the first controlled parcel and conclude once the last parcel is controlled. Additional constraints (19) and (20) eliminate model solutions that include unnecessary paths before and after the first and last controlled parcels respectively. Eliminating these solutions reduces the search space and helps the algorithm find the optimal solution, since a unique solution is identified given a set of controls. Figure 1 shows two different integer feasible solutions of the model that would represent the same solution to the original problem, since they have the same cells controlled. The one on the left is obtained when constraints (19) and (20) are not included. The one on the right is obtained when the additional constraints are included. A wandering path after the last controlled parcel appears when the constraints are not included. This kind of solutions arise because, when the wildfire is controlled, some parcels stop being threatened by the fire and can be visited by the brigade in any moment.

- *First control constraints.* These constraints force the first visited parcel of the suppression strategy to be controlled, since the way to arrive to the parcel is not considered:

$$z_i \geq s_i, \quad i = 1, \dots, n \quad (19)$$

- *Last control constraints.* These constraints force the last visited parcel of the suppression strategy to be controlled; avoiding solutions where the brigade continues the movement after finishing its job:

$$z_i \geq t_i, \quad i = 1, \dots, n \quad (20)$$

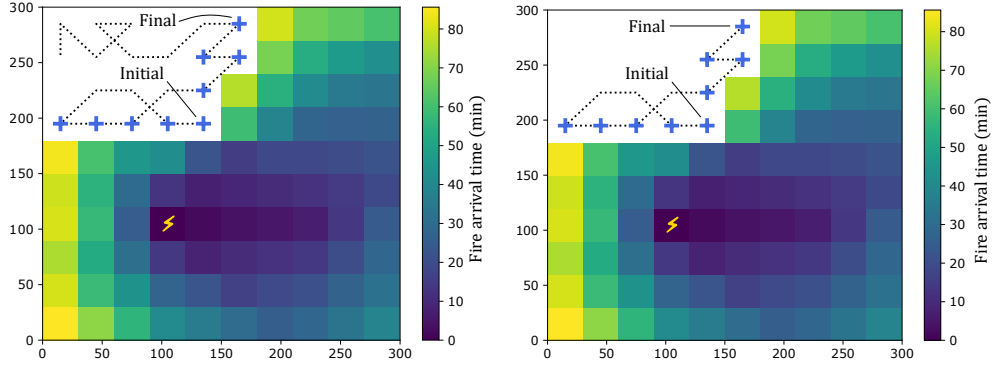


Fig. 1 On the left, a solution without constraints (19) and (20). On the right, a solution that includes them. Blue crosses represent controlled parcels and dashed line represents the brigade's path.

4.6 Criteria and objective function

The main goal of a wildfire suppression operation is to stop the fire spread, but there are several aspects to take into account. In this model, the main objective is to minimize the value of the burned parcels. Secondary aspects to be minimized are the cost of controlling parcels and the total time of operation.

- Value of burned parcels:

$$f_1(y) = \sum_i y_i d_i^f$$

where y_i is a binary variable representing if the parcel is burned and d_i^f is the value lost if the parcel i is burned.

- Value of controls:

$$f_2(z) = \sum_i z_i d_i^b$$

where z_i is a binary variable representing if the parcel i is controlled and d_i^b is the cost of controlling it.

- Total time-value of the operation:

$$f_3(x^b) = \sum_i x_i^b d^t$$

where x_i is the fire arrival time to parcel i and d^t the value of a unit time of operation.

Thus, this is a multicriteria problem which aim is to minimize $g(f_1, f_2, f_3)$. Several methodologies can be used to deal with it. In this case, scalarization and a two-stage scheme are used.

Scalarization

In order to obtain a compromise solution in a quick way, a scalarization scheme has been applied, combining the three criteria into a single objective function:

$$\min\{w^f f_1(y) + w^b f_2(z) + w^t f_3(x^b)\} = \min \left\{ \sum_{i=1}^n \left(w^f y_i d_i^f + w^b z_i d_i^b + w^t x_i^b d_i^t \right) \right\} \quad (21)$$

The weights (w^f, w^b, w^t) can be derived from the interaction with practitioners using historical and/or simulated cases as explained in Section 3. These weights are interpreted as directional scalar factors, where the increase of one of them leads to the improvement of the performance of the respective criterion. We can change the weights until getting the most suitable compromise solution for the decision-maker. A weak point of this approach is that the way the scalarizing function is built, as a weighted-sum, does not allow to find unsupported non-dominated solutions, which are typical of multi-objective mixed integer programming problems.

Two-stage scheme

Another way to deal with the conflicting criteria is to give them different levels of priority. Creating a multi-stage scheme allows optimizing the objectives sequentially, fixing a bound on previous criteria when optimizing subsequent ones.

In this case, a two-stage approach is used. The first objective function will be the value of the area burned, and the second stage would optimize the controlling costs plus the operational time.

$$\text{First stage} \rightarrow \min \{f_1(y)\} = \min \left\{ \sum_i y_i d_i^f \right\} \quad (22)$$

$$\text{Second stage} \rightarrow \min \{w^b f_2(z) + w^t f_3(x^b)\} = \min \left\{ \sum_i (w^b z_i d_i^b + w^t x_i^b d_i^t) \right\} \quad (23)$$

5 Computational experiments

Computational experiments are carried out to test the model on instances with different characteristics and using various solvers.

A set of cases has been created varying the landscape size (from a 5×5 grid to an 8×8 grid). For each size-case, 30² instances have been generated. In all the experiments performed to write this paper, the set of ignition points contains only one point. Systematically changing this ignition point gives rise to five groups of six instances with the same starting point within each group, but randomized fire spread parameters. Randomized values are based on parameters from FLAMMAP (Finney et al., 2006). These cases provide a diverse set to test the model in very different situations.

All the instances have been run on a computer with an Intel core i7 (9th gen.) 1.8 GHz processor with 16 GB ram. The model has been programmed in GAMS (version 37.1.0). A time limit of 3 h 30 minutes (≈ 13000 seconds) has been set for each run.

²This value has been adopted according to Coffin and Saltzman (2000), who state that for comparing the running times, samples in the range of 30-40 observations per treatment group should be sufficient.

This number, although is not suitable for operational purposes, is set to find integer optimal solutions for theoretical purposes, to check the performance of the model.

The model objective function used is the one from (21), using the scalarization. The value of the weights is based on Belval et al. (2015). A relation between w^f and w^b is established, where the cost of controlling a parcel is 0.1 times the lost value if that parcel is burned ($w^f = 0.1w^b$). In this case it is assumed that the value lost if a parcel i is burned (d_i^f) is the same for all parcels. The controlling costs (d_i^b) are also assumed to be equal for all parcels. The value of w^t is determined based on maximum brigade arrival times to weight and normalize the variable x_i^b in the objective function. The value is set to be $w^t = 10^{-5}w^f$. This relation between w^f and w^t is established such that w^f is sufficiently large with respect to w^t , in a way that a solution with less burned parcels is considered better than a solution with shorter operational times. The goal is to ensure that shortening the operational times is never made at the cost of letting one more parcel be burned.

5.1 Comparison with and without additional constraints

Additional constraints from Section 4.5, along with the reformulation from (17) to (18), have been set to speed up the process of finding a solution. Those constraints are a novelty compared to the model from Belval and Wei (2019) and improve the running times, as it can be seen in Table 1.

Average model runtime (s)	5x5	5x6	6x6	6x7	7x7	7x8	8x8
Base	548,17	3043,93	6371,78	6358,04	9128,24	9139,94	7467,65
Additional constraints	199,55	2248,93	6182,92	5337,03	8839,79	8518,04	7008,81
Average reduction (%)	45,26	56,98	25,64	51,67	28,56	30,56	41,73

Table 1 Average running times (in seconds) for the base model and the one including the additional constraints and the reformulation of equation (17) into equation (18). Average reductions are calculated comparing paired instances running times. Run using CPLEX (version 20.1.0.1).

From now on, all the results presented belong to the model including the additional constraints (19), (20) and the modified equation (18).

5.2 Comparison between solvers

To analyse the performance of the mixed integer linear programming model from Section 4, all the instances generated have been run with three different solvers (CPLEX version 20.1.0.1, GUROBI version 9.5.0 and XPRESS version v8.12.4). All the solvers have been used with their default parameters. The results, grouped by size-case can be seen in Figures 2 and 3, which are boxplots describing model runtimes and the relative gap reached within the time limit.

Table 2 displays performance metrics over the 30 instances within each size-case and using the three solvers. These metrics include minimum, maximum, mean, and standard deviation both for running times and gap. The last column (% solved) represents the percentage of instances for which the corresponding solver was able to

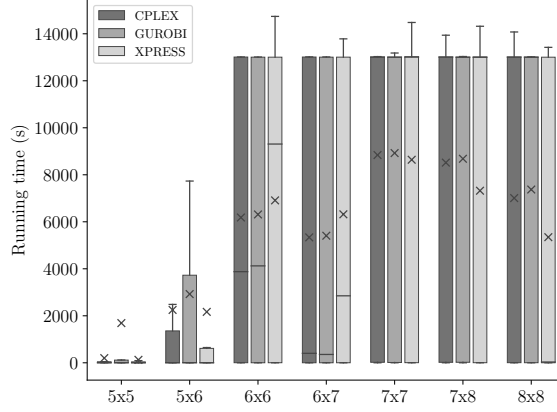


Fig. 2 Running time boxplot. Only instances where an integer feasible solution was found are included. Each landscape size includes 30 instances. $\times$ -like symbols represent average values. The outliers have not been plotted.

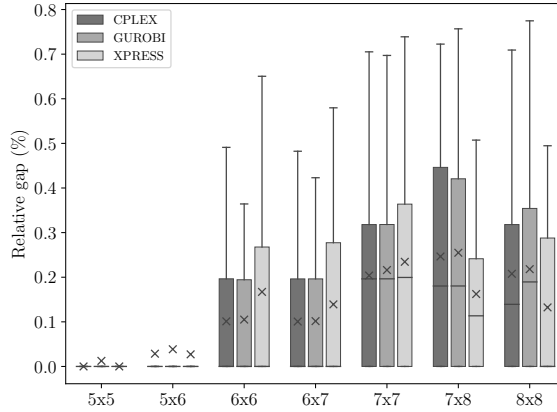


Fig. 3 Final gap boxplot. Only instances where an integer feasible solution was found are included. Each landscape size includes 30 instances. $\times$ -like symbols represent average values. The outliers have not been plotted.

find at least one feasible integer solution within each size-case. Rows in *italic* highlight those size-cases for which the corresponding solver was not able to find such a solution for some instances. Those instances are not included in the calculation of the performance metrics for that size-case, which could lead to wrong conclusions if it is not read jointly with the percentage of cases reaching an integer solution.

The first thing to note is that the bigger the landscape the longer the running times and the final MIP gaps. The model determines the best combination of parcels to control, and the number of possible combinations grows exponentially with the number of parcels to choose from. Average running times increase with landscape size, as well

Size	Solver	Running times (s)				GAP (%)				% solved
		min	max	mean	st. Dev.	min	max	mean	st. Dev.	
5×5	CPLEX	0.13	2176.81	199.55	467.88	0%	0%	0%	0%	100%
	GUROBI	0.20	13002.03	1689.57	3852.51	0%	20%	1%	5%	100%
	XPRESS	0.26	1031.34	126.33	261.06	0%	0%	0%	0%	100%
5×6	CPLEX	0.12	13011.10	2248.93	4535.61	0%	29%	3%	8%	100%
	GUROBI	0.17	13010.80	2924.19	5026.10	0%	39%	4%	9%	100%
	XPRESS	0.30	13005.09	2166.85	4464.88	0%	24%	3%	7%	100%
6×6	CPLEX	0.33	13012.53	6182.92	6272.79	0%	56%	10%	15%	100%
	GUROBI	0.48	13011.00	6312.18	6259.80	0%	54%	11%	16%	100%
	XPRESS	0.47	14739.34	6910.72	6430.55	0%	79%	17%	22%	100%
6×7	CPLEX	0.46	13016.12	5337.03	6232.66	0%	48%	10%	15%	100%
	GUROBI	0.55	13014.86	5407.21	6240.33	0%	64%	10%	16%	100%
	XPRESS	0.44	13784.12	6317.86	6523.30	0%	58%	14%	17%	100%
7×7	CPLEX	0.30	13029.39	8839.79	6061.54	0%	71%	20%	20%	100%
	GUROBI	0.51	13181.34	8924.08	6028.12	0%	70%	22%	21%	100%
	XPRESS	<i>0.53</i>	<i>14480.89</i>	<i>8640.25</i>	<i>6412.00</i>	<i>0%</i>	<i>74%</i>	<i>23%</i>	<i>22%</i>	<i>87%</i>
7×8	CPLEX	0.60	13939.15	8518.04	6179.76	0%	72%	25%	25%	100%
	GUROBI	0.64	13032.50	8681.99	6222.63	0%	76%	25%	28%	100%
	XPRESS	<i>0.93</i>	<i>14319.60</i>	<i>7322.20</i>	<i>6403.16</i>	<i>0%</i>	<i>62%</i>	<i>16%</i>	<i>20%</i>	<i>70%</i>
8×8	CPLEX	0.61	14076.71	7008.81	6612.80	0%	71%	21%	25%	100%
	GUROBI	0.87	13017.43	7372.85	6554.09	0%	77%	22%	25%	100%
	XPRESS	<i>0.77</i>	<i>13422.14</i>	<i>5345.69</i>	<i>6563.40</i>	<i>0%</i>	<i>49%</i>	<i>13%</i>	<i>18%</i>	<i>73%</i>

Table 2 Summary table of running times (in seconds) and gap for the different size-cases. Includes minimum, maximum, mean, and standard deviation both for running times and gap. The last column (% solved) represent the percentage of instances for which the corresponding solver was able to find at least one feasible integer solution within each size-case. Rows in italic highlight those size-cases for which the corresponding solver was not able to find such a solution for some instances.

as average MIP gaps. However, minimum running times stay stable in less than 1 second, reaching optimal solutions. This means that even though the landscape size increases, there are instances with certain characteristics that make them easy to solve. Most of them have the ignition point in a corner, which helps find a solution more easily because the border of the landscape is assumed to be non-flammable, so the wildfire is already partially enclosed.

It should be noted that the XPRESS solver is not able to find a feasible integer solution for a percentage of instances of the biggest cases. So, the runtime and gap values are obtained only for those instances reaching an integer solution, and should not be compared with the other solvers, which reach an integer feasible solution in all cases. This solver is not suitable to be used for this model.

Regarding the difference in performance between the other two solvers, CPLEX has proven to be slightly faster, and for the instances that it is not able to find the optimal solution, the gap is somewhat smaller than the one provided by GUROBI. As it can be seen in Figure 4, both solvers perform quite similar for most of the instances – differences among running times are around 0. Whereas the tails of the distribution show a slight asymmetry to the left, indicating that, in the cases where the running times differ importantly, there are more instances that are solved faster by CPLEX. When the difference among running times is larger than 1 h, there are 2 instances for

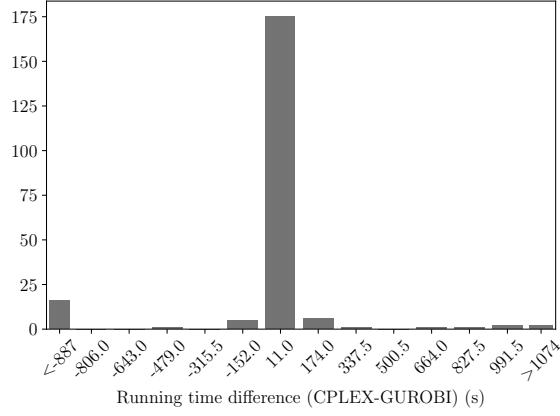


Fig. 4 Distribution of the differences in running times among CPLEX and GUROBI for each instance.

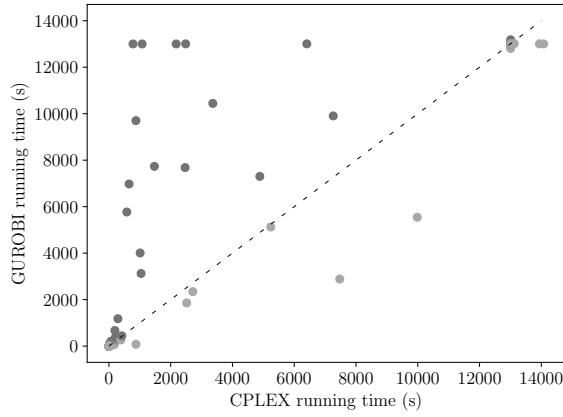


Fig. 5 Running times for all the 210 instances. CPLEX running times vs. GUROBI running times.

which GUROBI is faster, compared to the 11 instances for which CPLEX is faster. This trend can also be observed in Figure 5, where the running times for CPLEX and GUROBI are plotted against one another. This figure shows more observations (instances) in the upper left part of the graphic, indicating that for the same instance, GUROBI is slower than CPLEX.

Summarizing, the model has been tested using a test set of 210 instances and using three solvers, concluding that additional constraints are useful to eliminate feasible but sub-optimal solutions and that CPLEX is the most adequate solver for this model. However, large runtimes remain and therefore other solving strategies must be explored. Heuristics or decomposition techniques might be appropriate methodologies for this task.

5.3 Finding the optimal solution

Given the large running times, another experiment has been performed to detect the cause of the delay. Given the information from the 210 instances run with CPLEX, the time it takes to close the gap has been observed.

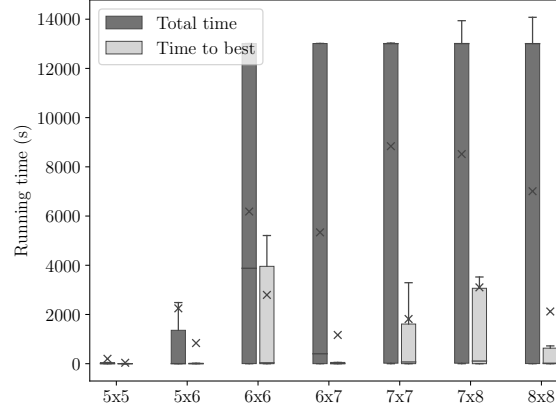


Fig. 6 Running time boxplot. All instances are included. *time to best* is the time to reach the optimal solution in the cases that finish with gap 0% and the time it takes to reach the best integer solution found for those that are not able to close the gap. $\times$ -like symbols represent average values. The outliers have not been plotted.

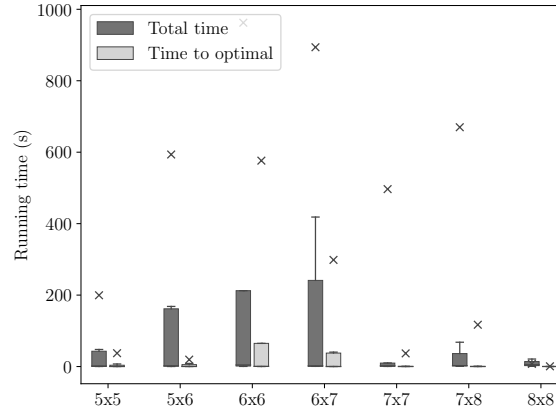


Fig. 7 Running time boxplot. Only instances that reach the optimal solution within the time-limit are included. $\times$ -like symbols represent average values. The outliers have not been plotted.

As it can be seen in Figures 6 and 7 the model spends most of the running time trying to close the gap. For several instances in the smallest landscape-sizes the integer

optimal solution is found within seconds and closing the gap takes several minutes. For the largest landscape-sizes the optimal solutions are more difficult to reach, so the comparison is made between the total running time and the time it takes to reach the best integer solution found. The model gets stuck improving the continuous bound for several hours without modifying the integer solution.

6 The two-stage approach

Given the conflicting criteria and provided that the area burned is the most important one, a two-stage approach is proposed, using the objective functions from (22) and (23).

Once the solver has been selected (CPLEX version 20.1.0.1), all the instances are run again using this approach. Given the large running times, two variants are proposed:

- Option 1: Once the first stage is run, the information passed to the second stage is just a bound on the area burned. The second stage optimizes the operational time and controlling costs creating new solutions as long as the area burned does not exceed the optimal one determined in the first stage.
- Option 2: Once the first stage is run, the information passed to the second stage is the exact location of the controls and which parcels end up burned. These act as parameters for the second stage, which only determines the optimal path to go through all the already fixed controls, minimizing the operational time.

Given that the first stage only pursues minimizing the area burned, multiple optimal solutions may arise, that will not be optimal for the second stage. Some of them are already discarded thanks to the additional constraints presented in section 4.5. These constraints prevent the brigade from wandering around before placing the first control or after the last one. However, the first stage is blind to ineffectual controls. To assure that all the controls are effective, they shall be next to a parcel that ends up burned. The next constraint is added to the model in its first stage:

$$z_i \leq \sum_{j \in A_i} y_j, \quad i = 1, \dots, n \quad (24)$$

It forces that every control should be placed next to at least one parcel that ends up burned. This constraint is of special interest for the *Option 2*, since controls are fixed from the first stage, and the second stage is not able to modify them.

The specifications of the machine on which the models are run are the same as explained in Section 5. In this case, the running time has to be shared between the two stages. A time limit of 3 h has been set for the first stage and at least 30 min is set for the second stage. If the first stage finishes within the 3 h limit time, the reminder can be used by the second stage.

As it can be seen, there are no significant differences between the areas burned between the three approaches or the controlling costs plus operational time (Figure 8 left and right respectively). We can conclude that the three approaches provide similar results. This reflects that the weights chosen for the scalarization are appropriate,

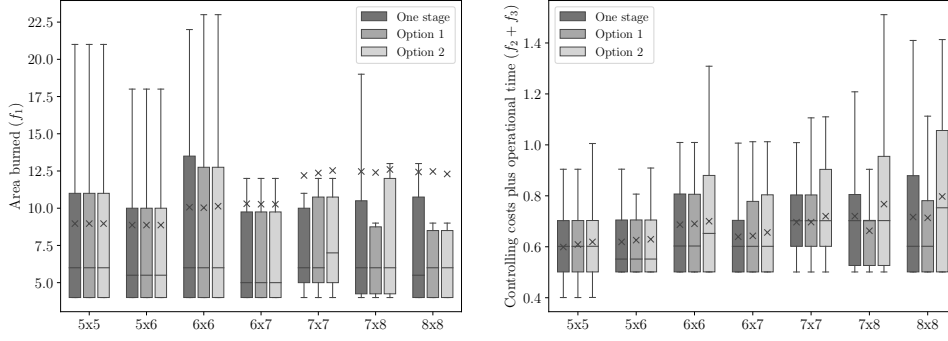


Fig. 8 Area burned f_1 (left) and controlling costs plus operational time $f_2 + f_3$ value (right) for each landscape-size. Outliers are not plotted and $\times$ -like symbols represent average values.

since they not allow for compensation between the criteria, providing the same results as the two-stage approach in which levels of priority are established.

Regarding running times, a slight reduction when using the two-stage approach Option 2 can be observed as shown in Figure 9. This trend is supported by the p-values calculated using the Wilcoxon signed-rank test, as shown in Table 3. In particular, the test is used with the null hypothesis (H_0) stating that Option 1 has lower running times than Option 2. Considering a 95% confidence level, we reject H_0 and so we can conclude that Option 2 is faster than Option 1. The non-parametric Wilcoxon signed-rank test is used because the data are paired and not normally distributed. P-values for cases 7×8 and 8×8 are not conclusive, since almost no instance finishes within the predefined time. Thus, the two-stage approach Option 2 is found to be the most suitable solving strategy, since the objective values are similar among the three approaches.

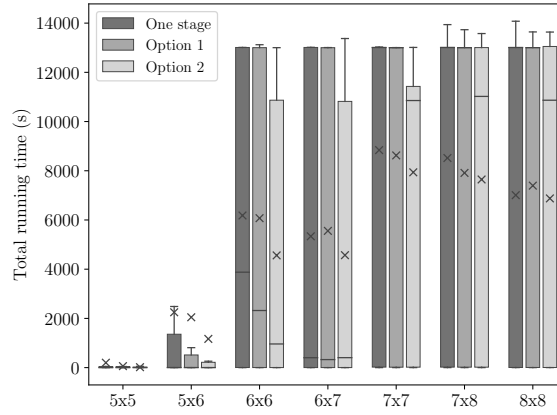


Fig. 9 Comparative boxplots for the total running time for each landscape-size. The outliers have not been plotted and $\times$ -like symbols represent average values.

	Option 1	Option 2	p-value
5×5	60.38	17.68	0.00001
5×6	2046.12	1167.33	0.00056
6×6	6073.19	4560.82	0.00008
6×7	5556.67	4569.62	0.00783
7×7	8623.84	7937.21	0.01750
7×8	7912.47	7645.75	0.20225
8×8	7395.74	6876.37	0.79193

Table 3 Average running times and comparative p-values for alternative models. The Wilcoxon signed-rank test is used to test H_0 : Option 1 has lower values than Option 2.

7 Case study

To validate the model, a case study has been created. It does not intend to be a real but a realistic case. Given the available raster information from the LANDFIRE program³, a 10×10 landscape is created, where a wildfire behaviour is simulated using FLAMMAP. The information provided by the simulator is then used to calculate the input parameters of the model, such as the time of fire spread from one parcel to another (a_{ij}^f) or the intensity it reaches in each parcel (b_{ij}), according to the methodology in Wei et al. (2011) and Belval et al. (2015).

The target landscape is a piece of terrain located in Weaverville, California 96093, U.S.A.. It is 300×300 m big, and it is discretized into a 10×10 grid.

This instance of the problem has been run in a computer with the characteristics described at the begining of section 5.

The two objective functions described in Section 4.6 are used as criteria. First, the model is run using the scalarization for the objective function. Second, the two-stage approach is used, implementing the Option 2 described in Section 6. For the scalarization method, a time limit of 24 h is set. In the case of the two-stage approach, 22 h are allowed to run the first stage, and 2 h for the second.

Figure 10 represents the outputs of the model for the case study using the scalarization of the objective function. These outputs include the wildfire behaviour description: the time it takes the fire to reach each parcel (colormap on the right of Figure 10, white for parcels not reached by the fire), the intensity the wildfire achieves in each parcel (colormap on the left of Figure 10) and the binding paths of the fire (black arrows on the left in Figure 10). The wildfire suppression strategy is determined by the blue crosses that represent controls' locations and the dashed line which is the path the brigade should follow to place the controls.

The starting point of the brigade's path is the parcel labelled as "initial" and the last one is labelled as "final". The initial location is near the wildfire starting point,

³<https://landfire.gov/lcp.php> accessed December 2022

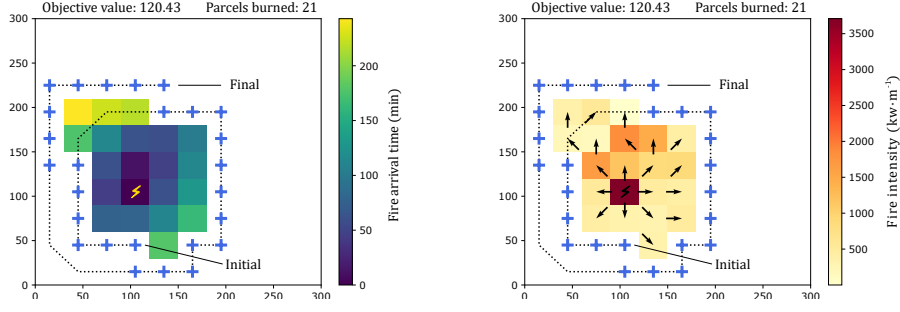


Fig. 10 Wildfire intensity, binding paths and arrival times for the case study solved via direct method. Crosses represent controlled parcels and the dotted line is the path the team should follow

because the parcels near the ignition location are the most urgent to control. If the fire spread is stopped when the wildfire is still small, the perimeter to be surrounded by controls will be smaller. It may catch the eye the fact that the brigade (dashed line) crosses some parcels that end up burned. This model allows the brigade to cross some parcels without controlling them, only to reach others, as long as the brigade leaves them before the wildfire arrives. An area of 21 parcels is burned. 79 parcels are saved, using 26 controls.

The solution presented in Figure 10 is the one the model achieved when the time limit was reached, since it kept improving the solution until the end. However, at the end of the 24 h run time, there was still a GAP of 39.7%.

Analogously, the results for the two-stages approach can be found in Figure 11.

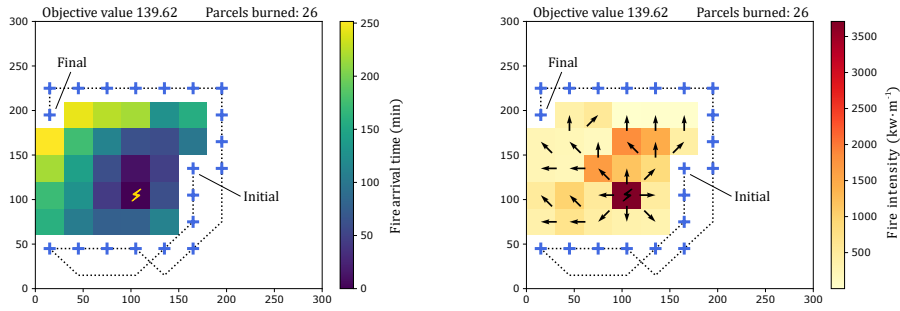


Fig. 11 Wildfire intensity, binding paths and arrival times for the case study solved via two-stage approach Option 2. Crosses represent controlled parcels and the dotted line is the path the team should follow

In this case 74 parcels are saved, using 20 controls. 26 parcels end up burned. The gap for the first stage is 48.5%, although the integer incumbent solution was found in 30 minutes running time – the rest of the 22 h time limit was dedicated to try closing the gap. The second stage gap is 0.68% certifying that, given the controls from the first stage, the second stage finds almost the optimal path between them.

As can be seen, better results are obtained when using the direct method for solving the problem. Mixing the value of the parcels along with the travel time in the

objective function helps the model find better solutions in a shorter time. Although minimizing the travel time is not the priority, shortening controlling and traveling times helps having more time available to reach endangered parcels sooner, allowing to create a smaller perimeter around the fire and surrounding it earlier, thus having less burned area. The two-stages approach is blind to operational time in the first stage, just trying to minimize burned area, but not having a hint of how important operational time is in this regard.

However, burned area and operational time are two different criteria, with different priorities and units of measure that should not be mixed in the same objective function. In the case of the computational study, the scalarization was used to be able to compare running times and gaps because it helps eliminating symmetries between solutions, but it is not recommended to analyze a real case, since it allows certain trade-off between objectives that are not on the same priority level.

7.1 Iterative approach

Running the model directly as made with the cases in the previous Subsection 5 can lead to no integer feasible solutions in limited time, or solutions with no real idea of how close they are to the optimal solution. Cases with more than 30 parcels are seldom solved to optimality in less than three hours runtime.

An iterative approach is proposed in this section, which improves the solution successively, leading to a more efficient way of searching the space of feasible solutions. It has been observed that limiting the controls' number leads to smaller running times. Moreover, the smaller the limit the shorter the running times. So, instead of searching for the optimal number of controls and their location from the beginning, the model looks for their optimal location given a small upper bound on their number (n), to obtain an initial integer feasible solution. Starting from zero (free burn), in each iteration the number of allowed controls is incremented and the model searches for a new solution, starting from the previous incumbent solution (x^*), since allowing to place more controls should lead to a solution at least as good as the previous one.

As explained in the previous section, mixing burned area and operational times in the objective function is not recommended. So, in this approach, the two-stages model is run in each iteration. A time limit of 10 minutes is set for each stage in each iteration and a gap of 3% is allowed since we do not look for the best solution in each step, but only for a reasonably good one to serve as a starting point for the next iteration.

The loop finishes when the objective function does not reach the specified gap. The last solution achieved is passed to the model with no bounds on the number of controls as an initial solution, to determine its optimality – or to find a better solution if possible.

This solving methodology helps find solutions more easily than solving the model directly, since it starts with a small number of controls, reducing the combinatorial complexity to spot an initial solution. Furthermore, it is more robust than a heuristic, since it provides an insight of how far the optimal solution is, since the MIP gap is analyzed in each step.

Algorithm 1 Iterative approach

```
 $n \leftarrow 0$  ▷ Set number of allowed controls to 0 (free burn)
Run the model. ▷ Fire arrival times on free burn. Very fast
 $x^* \leftarrow$  best incumbent solution ▷ Store solution with no controls
 $n \leftarrow 1$ 
while final gap  $\leq 3\%$  do
  initial solution  $\leftarrow x^*$  ▷ Initialize with best incumbent solution
  Run the two-stages model (time limit = 10 min)
   $x^* \leftarrow$  best incumbent solution ▷ Store solution with  $n$  controls
   $n \leftarrow n + 2$  ▷ Increment number of allowed controls
end while
initial solution  $\leftarrow x^*$ 
Run the model with no bounds on the number of controls
```

7.2 Results

In Table 4 it can be seen that the model is able to find solutions with bounds on the number of controls from one up to seventeen controls. Once seventeen controls are allowed, the model is able to find the minimum number of burned area, but not able to find the shortest path between the necessary controls, so the iterative phase finishes. The number of burned parcels decrease as the number of allowed controls increase. After the iterative phase, a last run of the model with no limits on the number of controls is performed to look for improvement, given a running time limit of 22h.

Controls limit	Controls placed	Burned parcels	Running time (s) 1st Stage	Gap (%) 1st Stage	Running time (s) 2nd Stage	Gap (%) 2nd Stage
1	1	99	1.57	2.86%	0.75	1.58%
3	3	96	0.599	2.97%	0.119	0.66%
5	5	91	3.846	0.00%	0.141	1.77%
7	7	84	7.496	0.00%	0.197	0.19%
9	9	76	5.231	1.77%	0.135	0.26%
11	11	71	600.966	2.10%	0.314	0.31%
13	13	59	600.321	2.68%	0.496	0.36%
15	15	43	600.261	2.20%	1.261	1.10%
17	17	37	600.265	2.72%	603.032	52.10%
Free	20	21	80024	38.64%	7854	1.00%

Table 4 Running times and gap for each iterative step with bound on the number of the controls

In Figure 12, the final solution from the model solved using the iterative approach is displayed. In the last run with no bounds on the number of controls the model places 20 controls, being the area burned 21.

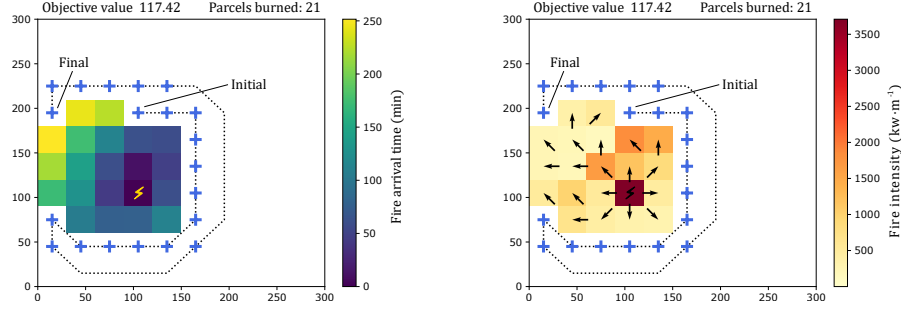


Fig. 12 Wildfire intensity, binding paths and arrival times for the case study, solved via iterative method. Crosses represent controlled parcels and the dotted line is the path the team should follow.

In comparison to the direct solving method, the iterative approach achieves equivalent results, placing six controls less than the solution provided by the direct solving method. The solution from the iterative approach takes advantage of the non-burnable border of the landscape, whereas in the direct method, the fire is completely surrounded by controls.

The final gap is 38.64% in the first phase, which is slightly better than the one obtained using the direct method. The iterative approach also achieves better results in terms of operational time plus controlling costs. Besides, it provides an insight of how far the optimal solution is, providing intermediate solutions, guaranteeing integer solutions for the model.

Summarizing, the iterative approach reaches better solutions in shorter times, and it can be stopped with an integer feasible solution at any time. However, in order to be useful for fire suppression in practice, runtimes are still too long and future research should be developed focusing on metaheuristic or matheuristic methods to solve the model.

8 Conclusions

The aim of this paper is to develop a wildfire suppression model, capable of providing suitable solutions for a detailed wildfire suppression strategy, with limited resources. The presented model includes constraints representing the fire spread and its interaction with the wildfire suppression strategy defining fire arrival times, which in turn limit the reachability of the parcels to be crossed or controlled. A detailed wildfire suppression strategy is determined in terms of the controls, their location and timing, along with the shortest path a brigade should follow to place those controls, considering safety concerns.

The model includes features from the previously developed MTT methodology to create a more realistic model representing the interaction of wildfire behaviour

and wildfire suppression strategy. Being the model similar to the one in [Belval and Wei \(2019\)](#), it would have been interesting to compare the results of the presented model with those by [Belval and Wei \(2019\)](#), but they are not comparable, due to the stochastic formulation and the way the control time of parcels is calculated in that model.

Three criteria are considered to determine the quality of a solution – the value of the area burned, the value of the controls and the time of operation. All of them to be minimized. They are conflicting objectives, so a multiple criteria approach is needed to solve it. Scalarization has been chosen to compare different modelling and solving strategies, and a two-stage approach to prioritize minimizing the value of the area burned over the other objectives.

A computational study has been carried out to give an overview of the model’s performance and limitations, such as the long runtimes and big gaps, as well as difficulties to find a feasible integer solution with some solvers and grid sizes greater than 7×7 . The model is validated on a realistic 10×10 -sized case study, and an iterative methodology is presented to obtain good solutions for large instances, based on solving models with limited number of controls. Despite it is not able to solve the case study to optimality, it provides insight into optimal solutions for a reduced number of controls in a very short time.

Summarizing, the contributions of this work encompass a novel formulation of the problem, developing enhancements driven by additional constraints and two levels for prioritizing the criteria. Additionally, a computational study is conducted, including an iterative approach for dealing with large instances.

One of the limitations of the model is the assumption of each parcel only being crossed once. This is due to the necessity of tracking the visiting instant, for safety constraints. However, crossing parcels represent roads or paths that might and should be used more than once as long as it is safe. Furthermore, an improved version of the model should be able to represent the cooperation between several brigades. Also, the fire intensity could be accounted for developing strategies avoiding engagement in dangerous points where the fire intensity is too high. Determining strategies that include overlapping paths, multiple brigades or fire intensity as a safety parameter is out of the scope of this paper, but they are key elements for future enhancement.

The simplifying assumptions made, and in particular, the reduced size of the instances are useful to establish a starting point for studying the problem. This paper has formulated an exact model to solve it, that helps identifying its structure and clearly defining its constraints. However, the already large model runtimes are expected to increase if these simplifications are removed. The carried computational study has certainly demonstrated that more powerful tools are needed to deal with it, so alternative solving methodologies must be explored to make the model more useful for practitioners. These methodologies might include decomposition techniques or heuristics.

Conflict of interest. The authors declare no conflict of interest.

References

- Bayham, J., & Yoder, J.K. (2020). Resource allocation under fire. *Land Economics*, 96(1), 92–110, <https://doi.org/10.3368/le.96.1.92>
- Belval, E.J., Stonesifer, C.S., Calkin, D.E. (2020). Fire suppression resource scarcity: Current metrics and future performance indicators. *Forests*, 11(2), 217, <https://doi.org/10.3390/f11020217>
- Belval, E.J., & Wei, Y. (2019). Modeling Ground Firefighting Resource Activities to Manage Risk Given Uncertain Weather. *Forests*, 10(12), 1077, <https://doi.org/10.3390/f10121077>
- 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,
- Bodaghi, B., Palaneeswaran, E., Shahparvari, S., Mohammadi, M. (2020). Probabilistic allocation and scheduling of multiple resources for emergency operations; a Victorian bushfire case study. *Computers, Environment and Urban Systems*, 81, 101479, <https://doi.org/10.1016/j.compenvurbsys.2020.101479>
- Chow, J.Y., & Regan, A.C. (2011). Resource location and relocation models with rolling horizon forecasting for wildland fire planning. *INFOR*, 49(1), 31–43, <https://doi.org/10.3138/infor.49.1.031>
- Coffin, M., & Saltzman, M.J. (2000). Statistical analysis of computational tests of algorithms and heuristics. *INFORMS J. Comput.*, 12, 24–44,
- Curt, T., & Fréjaville, T. (2017). Wildfire policy in mediterranean france: How far is it efficient and sustainable?: Wildfire policy in mediterranean france. *Risk Analysis*, 38(3), 472–488, <https://doi.org/10.1111/risa.12855>
- Donovan, G.H., & Rideout, D.B. (2003). An integer programming model to optimize resource allocation for wildfire containment. *Forest Science*, 49(2), 331–335, <https://doi.org/10.1093/forestscience/49.2.331>

- 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, <https://doi.org/10.1007/s10342-015-0933-8>
- Finney, M.A., et al. (2006). An overview of flammap fire modeling capabilities. *Fuels management—how to measure success: conference proceedings* (Vol. 28, p. 30).
- Granda, B., León, J., Vitoriano, B., Hearne, J. (2023). Decision support models and methodologies for fire suppression. *Fire*, 6(2), 37, <https://doi.org/10.3390/fire6020037>
- Homchaudhuri, B., Kumar, M., Cohen, K. (2013). Genetic algorithm based simulation-optimization for fighting wildfires. *International Journal of Computational Methods*, 10(6), 1350035, <https://doi.org/10.1142/S0219876213500357>
- Hu, X., & Ntaimo, L. (2009). Integrated simulation and optimization for wildfire containment. *ACM Transactions on Modeling and Computer Simulation*, 19(4), 1–29, <https://doi.org/10.1145/1596519.1596524>
- Ingalsbee, T., & Raja, U. (2015). The Rising Costs of Wildfire Suppression and the Case for Ecological Fire Use. D.A. DellaSala & C.T. Hanson (Eds.), *The ecological importance of mixed-severity fires* (pp. 348–371). Elsevier Inc.
- Mendes, A.B., & Alvelos, F.P. (2023). Iterated local search for the placement of wildland fire suppression resources. *European Journal of Operational Research*, 304(3), 887–900, <https://doi.org/10.1016/j.ejor.2022.04.037>
- Ríos-Mercado, R.Z., González-Tagle, M.A., Díaz-Romero, M.A., Rodríguez-y Silva, F. (2020). An Integrated Framework for Control and Management of Forest Fires Using Potential Fire Risk Indices. *Preprint PISIS-RR20-02, Graduate Program in Systems Engineering, UANL*(September), ,
- Rodríguez-Veiga, J., Guinzo Villamayor, M.J., Casas Mendez, B. (2018). An integer linear programming model to select and temporally allocate resources for fighting forest fires. *Forests*, 9(10), 1–18, <https://doi.org/10.3390/f9100583>
- Shahidi, A., Ramezani, R., Shahparvari, S. (2022). A greedy heuristic algorithm to solve a vrp-based model for planning and coordinating multiple resources in

emergency response to bushfires. *Scientia Iranica*, , <https://doi.org/10.24200/sci.2022.57476.5258>

- Shi, G., Yan, H., Zhang, W., Dodson, J., Heijnis, H., Burrows, M. (2021). Rapid warming has resulted in more wildfires in northeastern Australia. *Science of the Total Environment*, 771, 144888, <https://doi.org/10.1016/j.scitotenv.2020.144888>
- Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35, 254-265, <https://doi.org/10.1287/opre.35.2.254>
- Suarez, D., Gomez, C., Medaglia, A., Akhavan-Tabatabaei, R. (2016). *A stochastic programming approach for wildfire suppression : Pre-positioning and distribution of resources under uncertainty* (Unpublished doctoral dissertation). Universidad de los Andes, Colombia.
- Tomasini, R., & van Wassenhove, L. (2009). *Humanitarian Logistics*. Palgrave Macmillan.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. SIAM.
- Wei, Y., Belval, E.J., Thompson, M.P., Calkin, D.E., Stonesifer, C.S. (2017). A simulation and optimisation procedure to model daily suppression resource transfers during a fire season in Colorado. *International Journal of Wildland Fire*, 26(7), 630–641, <https://doi.org/10.1071/WF16073>
- 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(October), 1–29, <https://doi.org/10.1111/nrm.12295>
- 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, <https://doi.org/10.3390/f10040311>

- Yang, Z., Guo, L., Yang, Z. (2019). Emergency logistics for wildfire suppression based on forecasted disaster evolution. *Annals of Operations Research*, 283(1-2), 917–937, <https://doi.org/10.1007/s10479-017-2598-9>
- Zeferino, J.A. (2020). Optimizing the location of aerial resources to combat wildfires: a case study of Portugal. *Natural Hazards*, 100(3), 1195–1213, <https://doi.org/10.1007/s11069-020-03856-6>
- Zylstra, P.J. (2018). Flammability dynamics in the Australian Alps. *Austral Ecology*, 43(5), 578–591, <https://doi.org/10.1111/aec.12594>