Fuel Management Operations Planning in Fire Management: a Bilevel Optimisation Approach

Federico Liberatore[a,*], Javier León[b], John Hearne[c], Begoña Vitoriano[b]

[a] *School of Computer Science & Informatics, Queens Building, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, United Kingdom. UC3M-Santander Big Data Institute (IBiDat), Charles III University of Madrid, Madrid, Spain*
[b] *Institute of Interdisciplinary Mathematics and Department of Statistics and Operational Research, Complutense University of Madrid, Madrid, Spain*
[c] *School of Science, RMIT University, Melbourne, Australia*

Abstract

Elevated fuel loads represent a wildfire hazard in a landscape. Reducing fuel load is one mitigation strategy commonly employed to decrease the severity and impact of wildfires. The planning of such fuel management operations, however, represents a complicated decision problem, which includes multiple sources of uncertainty. In this paper, a problem for fuel treatment planning is presented, formulated, and solved. The optimisation model identifies the best subset of units in the landscape to be treated to minimise the impact of the worst-case wildfire. Due to its size, which would make it intractable for realistic instances, an ad hoc exact solution algorithm has been devised. Extensive computational testing on randomly generated instances illustrates that the proposed approach is very successful at solving the problem. Finally, the algorithm is applied to a case study on a landscape in Andalusia, Spain, which shows the capabilities of the proposed approach in addressing a real-world problem.

Keywords: Fire Management, Fuel Management, Wildfire, Attacker-Defender Model, Operational Research

1. Introduction

Every year, hundreds of thousands of square kilometres of forests and other types of land cover burn due to wildfires. The impact of these events is catastrophic, with significant economic and ecological losses, and often, human casu-

*Corresponding author

Email addresses: liberatoref@cardiff.ac.uk (Federico Liberatore), javileon@ucm.es (Javier León), john.hearne@rmit.edu.au (John Hearne), bvitoriano@mat.ucm.es (Begoña Vitoriano)

Article published in Safety Science, Volume 137, May 2021, 105181. Accepted manuscript accessible at https://doi.org/10.1016/j.ssci.2021.105181.

alties. Also, major wildfires have long-lasting effects, as the emissions generated can affect the climate and air quality in the local area as well as in neighbouring ones. The 2019-20 Australian bushfire crisis is the most recent and tragic event of such kind. During this crisis, wildfires burned more than 10.3 million hectares (a territory comparable to the island of Britain) (BBC News, 2020), killed an estimated one billion animals (The University of Sydney, 2020), and emitted 306 million tonnes of carbon dioxide, among others substances and toxic pollutants (Lee, 2020). Globally, wildfire-related destruction is a problem that appears to be worsening. This upward trend seems set to continue due to rising temperatures and altered weather conditions associated with climate change. Prevention and expeditious and effective response to wildfires can help to reduce the impact of these disruptive events.

Fire management is the process of planning, preventing and fighting fires to protect people, property and natural resources. The different tasks that comprise fire management can be roughly grouped into the following categories: fuel management; fire prevention; detection; suppression resource acquisition, deployment, dispatch and use; and large fire management (Martell, 2015). These tasks present several intrinsic complexities and fire managers operate in a very challenging decision environment characterised by a high degree of uncertainty (Minas et al., 2012). The most unpredictable factors in this context are weather forecasts, the performance of suppression resources, and fire behaviour, spread and effects (Pacheco et al., 2015). As in other fields sharing similar characteristics (e.g., disaster management, policing, public safety and security), the use of optimisation models often results in a reduction of the level of subjectivity present in the decision-making process and an improvement in the quality of the decisions (Camacho-Collados and Liberatore, 2015). The need for systematic, analytical tools to support processes, decision-making and planning in fire management has been recognized among academics and practitioners alike (Martell, 2015; The England & Wales Wildfire Forum).

This paper focuses on the task of fuel management. Fuel management is the planned manipulation of the amount, composition, and structure of the biomass within wildland ecosystems to modify potential fire behaviour and effects (Husari et al., 2006). One of the strategies adopted is isolation, which seeks to decrease the area burned by compartmentalising fires (Fernandes, 2013). This can be achieved by using different treatments, such as mechanical clearing, prescribed burning and controlled grazing (Marino et al., 2014). It is a common practice among fuel managers to look at the state of the landscape and plan for treatments every year, before the start of the fire season.

In this paper, an optimisation model to support fuel management operations is proposed. Given a landscape divided into units and their attributes (i.e., size, hazardousness, and connections to other units), a defender (i.e., the fuel manager) wants to identify the best subset of units to be treated to minimise the impact of wildfires. A budget limits the total area that the defender can treat. It is assumed that a treated unit cannot suffer fire again during the planning period. Uncertainty in ignition locations and the subsequent wildfire behaviour is modelled by introducing an attacker (i.e., the fire) with complete information on the landscape and the defender's actions. The attacker chooses a number of hazardous units as wildfire ignition points, with the objective of destroying as much area as possible. It is assumed that the wildfire expands from one burning unit to all the neighbouring hazardous units, thus burning all the hazardous units connected to an ignition point. Given that a treated unit is not affected by fire, the defender must use the budget available to fragment large connected hazardous areas and, consequently, isolate the ignition points as much as possible. Since the attacker acts intelligently, the model minimises the impact of the worst-case outcome, that is, the worst possible wildfire. This conservative approach ensures that a fire cannot spread above the objective value, identifying treatments that result in relatively small and controllable wildfires. Thus, it identifies fuel treatment strategies that are effective against both natural and anthropogenic wildfires. A graphical representation of a sample problem instance and its solution is given in Figure 1.

The contribution of this paper to the literature is three-fold. The first one is the formulation of a problem for developing optimal fuel management plans to counter wildfires. The second contribution is the development of an exact method to solve the problem. Finally, the last contribution is a novel dataset that constitutes a challenging testbed for this and related problems.

The paper is organised as follows. Section 2 reviews the literature relevant to the context studied. Section 3 presents the problem addressed in detail and two formulations.The algorithm devised to solve the problem to optimality is the subject of Section 4. The proposed algorithm is tested on randomly generated problem instances and the results are analysed in Section 5. In Section 6 the methodology is then applied to a real-world case study to illustrate the applicability and usefulness of the methodology devised. Next, Section 7 provides practitioners with practical guidelines on how to apply and use the model. Finally, the paper concludes with some insights and guidelines for future research.

2. Literature Review

The model proposed in this research aims to reduce the impact of wildfires by fragmenting large hazardous territories into smaller disconnected areas. This is the rationale of a number of models that have been previously presented in the literature. To the best of the authors' knowledge, the first model along this line is the one by Wei et al. (2008). The authors compute a fire risk distribution map using fire simulations. This is then used by an optimisation model to locate fuel treatments with the objective of breaking patterns of fire risk accumulation following the wind direction. Minas, Hearne, and Martell improve on this approach in a couple of ways. In Minas et al. (2015) a model that integrates fuel management decisions and fire suppression preparedness operations is proposed, while in Minas et al. (2014) they introduce a temporal dimension, analysing the strategic implications of multi-period fuel treatment plans. Rachmawati et al. (2018) and León et al. (2019) build on the latter contribution by incorporating objectives and constraints dealing with habitat goals. In fact, both their models aim at obtaining a mixture of vegetation composition in the landscape for environmental reasons, while decreasing the connectivity of dangerous areas. Matsypura et al. (2018) measure fuel accumulation using Olson curves (Olson, 1963) rather than a linear function, as in the previously mentioned papers. Alternative formulations are discussed in the reviews by Minas et al. (2012) and Gillen et al. (2017).

As mentioned in the introduction, uncertainty is an important element in fire management, due to the numerous unpredictable factors that have to be taken into account. Therefore, some authors have proposed models that incorporate and address randomness, following different strategies for its inclusion into optimisation models. Wei and Long (2014) assign to each unit of the landscape a probability of fire ignition. Therefore, the objective is the minimisation of the expected loss deriving from a single wildfire which could occur at any unit. In the model, the wildfire spreads according to the minimum travel time (MTT) algorithm (Finney, 2002), which basically consists of growing the fire along the shortest paths originating from the ignition point. Also, the wildfire is assumed to be contained after a fixed amount of time. This assumption is clearly unrealistic, as the time required to control a fire depends on multiple factors, including its size and intensity, the weather, and the resources employed to extinguish it. Kabli et al. (2015) propose a classical two-stage stochastic programming (SP) model, where the first stage decision concerns which treatments to apply, and the second stage only evaluates the outcome. It is the opinion of the authors that this approach has several shortcomings. Firstly, it does not consider that fire spreads and can become uncontrollable. Secondly, both the treatment resources and the consequences of the fire are evaluated in terms of cost, which can create a trade-off between them and is completely contrary to the rationale of disaster management. A wildfire can have repercussions that can be hard to monetarily quantify, such as, loss of biodiversity and human lives. Thirdly, the definition of the scenarios requires detailed probability information on multiple sources of uncertainty, such as weather forecasts and fuel levels. In general, more complex models allow for more realistic decisions. However, in practice, it is usually difficult to obtain enough reliable, high-quality, historical data to accurately estimate the parameters' probability distributions. Regarding the article under analysis, very little information is given by the authors on how this data can be obtained and how the scenarios are generated. Finally, the methodology is tested only on a single case study consisting of 15 locations and six scenarios.

Robust optimisation (RO) offers a different approach from SP to tackling uncertainty. RO does not require the probability distribution of the random parameters, which just need to be defined by an uncertainty set (Ben-Tal and Nemirovski, 2002). This results in models that are less sensitive to data perturbations than deterministic models, require less data than SP programs and are computationally tractable (Gorissen et al., 2015). Minimax models are a paradigm of RO having the objective of minimising the impact of the worst-case outcome (Snyder, 2006), that is, improving as much as possible the effect of the most severe possible outcome that can reasonably be projected to occur in a given situation. In the field of game theory and critical infrastructure, minimax models are often referred to as attacker-defender models.

The problem proposed in this research belongs to the family of attacker-defender models, a special type of Stackelberg game (Von Stackelberg, 2010) in which the objective of the defender is to impair the objective of the defender by allocating limited protection resources to elements of a system that the attacker wants to damage as much as possible. Due to their hierarchical structure, such models are normally represented in the literature as bilevel or multi-level optimisation programs. To the best of our knowledge, the concept of protecting elements of a system against attacks was originally discussed by Salmeron et al. (2004) in the context of electrical power grids, although no formal model is given by the authors. Some of the first models in the field have dealt with the analysis of vulnerabilities in electric power grids, subways, airports, and other critical public infrastructure (Brown et al., 2005), the allocation of protection resources to elements of water supply networks against physical attacks (Qiao et al., 2007), and the definition of protection strategies for supply chains (Church and Scaparra, 2007; Scaparra and Church, 2008; Liberatore et al., 2011; Liberatore and Scaparra, 2011; Liberatore, 2012). The first contributions in the literature that explicitly consider natural disasters were the game-theoretical model by Zhuang and Bier (2007), and the protection of facility networks against ripple-type disruptions (e.g., earthquakes and floods) by Liberatore et al. (2012).

Attacker-defender models have also been applied to fuel management. Rashidi et al. (2018b) propose a model to counter against pyro-terror attacks through fuel treatments, called the pyro-terrorism mitigation problem (PTMP). Pyro-terrorism consists of large-scale human-caused wildfires for political or religious purposes. Despite the focus on intentional attacks, PTMP presents several elements in common with the one proposed in this paper. Both models represent the landscape as a graph and the defender has a limited budget that determines the number of nodes that can be treated. However, PTMP only contemplates a single attack and the landscape must be represented as a grid. The model presented in this paper improves on it by admitting landscapes of any topology, which makes it more realistic and greatly increases its applicability, and by allowing for multiple ignition points. However, the main difference between the two approaches lies in the representation of wildfire behaviour. PTMP builds upon the representation by Wei and Long (2014) and, therefore, it suffers from the same limitations. Namely, the fire-spreading model underlying PTMP assumes that the fire has a maximum fire duration after which it is controlled, regardless of its intensity, size, and other characteristics. On the other hand, the model proposed in this paper simply assumes that the fire affects all the hazardous territory that it can reach from the ignition point. This approach is conservative and leads to solutions that are more robust than PTMP, as they tackle the worst possible outcome.

Although not directly relevant to this review, it is interesting to mention that, in a subsequent article, Rashidi et al. (2018a) propose the vulnerability assessment of the initial attack problem (VAIAP), which extends PTMP in multiple ways and shifts the focus from fuel management to suppression. In

VAIAP, the attacker can locate multiple ignition points, while the defender takes two types of allocation decision: the pre-attack location of suppression resources to fire stations, and the post-attack dispatch of these resources to control the fires. Also, VAIAP overcomes the greatest shortcoming of its predecessor and, instead of assuming that the fire can be controlled after a fixed amount of time regardless of its size, it calculates the number of wildfires that escape containment and tries to minimise it. This is an improvement and an interesting angle that is achieved by estimating the length of the fire line at the time of intervention for each of the original ignition points. However, VAIAP assumes that the fires do not interact or merge. This assumption is simplistic as wildfires do interact and may exacerbate each other, leading to an underestimation of their effects.

3. Model Formulation

The following section provides technical details of the methodology. Readers only interested in the practical outcomes of this work may omit this section.

The problem studied and solved is a Stackelberg game between two players: a defender (i.e., the fuel manager) and an attacker (i.e., the fire). The game is played on a graph (i.e., a landscape divided into burn units). Two nodes are connected by an edge if a wildfire could spread from one node to the other. Each node is characterised by its area. Also, some nodes are hazardous, meaning that they could suffer fire. These nodes form clusters in the graph, i.e., subgraphs of connected hazardous nodes. Figure 2 illustrates an example.

Firstly, the defender chooses a subset of nodes according to a treatment budget. These nodes are protected against fire and, therefore, cannot suffer fire. Next, the attacker chooses a subset of nodes to strike, given a limited capacity. If a node chosen by the attacker is hazardous, then a fire starts. As a result, the whole cluster to which the struck node belongs to is burned. The value of the solution is the total burned area. The objective of the attacker is to maximize the value, while the objective of the defender is to minimize it. This problem translates naturally into a bilevel program, which is presented in the following.

3.1. Bilevel Formulation

Sets

- N, set of nodes, indexed by i and j.
- $\bar{N}$, subset of hazardous nodes.
- E, set of directed edges, indexed by (i, j).
- $\bar{E}$, subset of directed edges connecting hazardous nodes.

Parameters

- $s_i > 0$, area of node i.
- M_i, cardinality (i.e., number of nodes) of the cluster to which i belongs to.
- $S > 0$, defender's treatment budget.
- $B \in \mathbb{N}$, attacker's capacity.

Variables

The decision variables for the attacker and the defender are:

- $x_i = \begin{cases} 1 & \text{if node } i \text{ is treated by the defender,} \\ 0 & \text{otherwise.} \end{cases}$
- $y_i = \begin{cases} 1 & \text{if node } i \text{ is struck by the attacker to start a fire,} \\ 0 & \text{otherwise.} \end{cases}$

The formulation also makes use of support variables that represent the behaviour of the fire resulting from the actions of the defender and the attacker:

- $burn_i = \begin{cases} 1 & \text{if node } i \text{ is burned by fire,} \\ 0 & \text{otherwise.} \end{cases}$
- $flow_{ij} \geq 0$, flow representing the fire spreading from nodes i to j.
- $supply_i \geq 0$, flow supply at node i.

Formulation

Defender Problem [DP]:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \tag{1}$$
$$s.t. \quad \sum_{i \in \bar{N}} s_i \cdot x_i \leq S \tag{2}$$
$$x_i \in \{0, 1\} \qquad \forall i \in \bar{N} \tag{3}$$

The objective of the defender (1) is to minimise the impact of the fire resulting from the attacker's actions by optimally choosing hazardous nodes to treat, according to the available budget (2). Constraints (3) state that no partial treatment of the nodes is possible.

Attacker Problem [AP]:

$$f(\mathbf{x}) = \max_{\mathbf{y}} \quad g(\mathbf{x}, \mathbf{y}) \tag{4}$$
$$s.t. \quad \sum_{i \in \bar{N}} y_i \leq B \tag{5}$$
$$y_i \leq 1 - x_i \qquad \forall i \in \bar{N} \tag{6}$$
$$y_i \in \{0, 1\} \qquad \forall i \in \bar{N} \tag{7}$$

In contrast with DP, the objective of the attacker (4) is to burn as much area as possible by identifying the worst-case set of B hazardous nodes (5) to strike, chosen among those which are not treated by the defender (6). Finally, a node cannot be partially struck (7).

System Model [SM]:

$$g(\mathbf{x},\mathbf{y}) = \textstyle\sum_{i\in\bar{N}} s_i \cdot burn_i \tag{8}$$

$$burn_i \geq y_i \qquad \forall i \in \bar{N} \tag{9}$$

$$burn_j \geq burn_i - x_j \qquad \forall (i,j) \in \bar{E} \tag{10}$$

$$burn_i \leq 1 - x_i \qquad \forall i \in \bar{N} \tag{11}$$

$$burn_i \leq y_i + \textstyle\sum_{j:(j,i)\in\bar{E}} burn_j \qquad \forall i \in \bar{N} \tag{12}$$

$$supply_i \leq M_i \cdot y_i \qquad \forall i \in \bar{N} \tag{13}$$

$$supply_i + \textstyle\sum_{j:(j,i)\in\bar{E}} flow_{ji} = \textstyle\sum_{j:(i,j)\in\bar{E}} flow_{ij} + burn_i \qquad \forall i \in \bar{N} \tag{14}$$

$$burn_i \in \{0,1\} \qquad \forall i \in \bar{N} \tag{15}$$

$$flow_{ij} \geq 0 \qquad \forall (i,j) \in \bar{E} \tag{16}$$

$$supply_i \geq 0 \qquad \forall i \in \bar{N} \tag{17}$$

SM defines the behaviour of the fire on the landscape based on the actions of the players and computes the score $g(\mathbf{x},\mathbf{y})$ as the total burned area (8). The first two groups of constraints force a hazardous node to burn if it is struck by the attacker (9) or if it is connected to a burned node (10). Next, the following two groups of constraints state that a node can burn only if it has not been treated by the defender (11) and if it is struck by the attacker or it is connected to another burned node (12). The constraints presented so far are necessary but not sufficient to properly model the behaviour of the fire. In fact, it allows for clusters that spontaneously combust without any attacker's action, in a way similar to forming subtours in the travelling salesmen problem. Therefore, the constraints (13)-(14) are introduced to prevent this undesirable behaviour by enforcing flow conditions: each burned node requires a unit of flow (14) that can only be emitted by a node struck by the attacker (13). Therefore, nodes belonging to clusters unaffected by the attacker cannot burn.

Model Analysis

It is well-known that bilevel programs are hard problems due to their inherent non-convexity and non-differentiability (Bard and Falk, 1982). Even the simplest case, the linear bilevel program (LBP), has been shown to be strongly NP-hard (Bard, 2013) and it has been proven that merely evaluating a solution for optimality is also a NP-hard task (Vicente et al., 1994). A conventional method to solve a LBP is to replace the lower level problem with optimality conditions (i.e., by its Karush–Kuhn–Tucker, KKT, conditions).

The problem studied in this paper is a mixed integer bilevel programming (MIBP) problem, as it presents binary variables in both the upper and lower problems (i.e., DP and AP, respectively). MIBP problems are even more difficult to solve than standard LBPs. In fact, they generally cannot be tackled using conventional methods, such as the KKT approach mentioned above.

In the bilevel programming model, DP and AP have the same objective function, yet with opposite optimisation directions. If AP had the integrality property, then the integrality condition on variables $\mathbf{y}$ could be relaxed and the solutions would still be integer. Therefore, for a given $\mathbf{x}$, we could consider the dual attacker problem (DAP), and embed it into DP, by the strong duality theorem in linear programming (LP) (Matousek and Gärtner, 2007). DAP would present non-linear complementarity constraints that could be linearised using standard linearisation techniques (Glover, 1975; Kettani and Oral, 1990; Chang, 2000; Adams and Forrester, 2005; Sherali and Adams, 2013), resulting in a mixed integer linear programming (MILP) problem.

Unfortunately, this is not the case for the problem considered. In fact, a program has the integrality property if all the right-hand side values are integer and if the constraint coefficients matrix is totally unimodular. It can be easily seen, however, that the constraint coefficients matrix of AP is not totally unimodular. For example, the M_i coefficient of variable y_i in constraints (13) can be different from -1, 0, or 1. As a consequence, the dualisation approach is not a viable option and a more specific methodology must be devised to solve it.

Despite not having the integrality property, AP is still trivial for a specific defender strategy. Given a fixed $\mathbf{x}$, AP can be formulated as follows. Please note that the following notation supersedes any previous definition.

- C, set of hazardous clusters, indexed by c.
- $B \in \mathbb{N}$, number of clusters chosen by the attacker (i.e., attacker's capacity).
- $s_c > 0$, area of cluster $c \in C$.
- $y_c = \begin{cases} 1 & \text{if the cluster } c \in C \text{ is chosen by the attacker,} \\ 0 & \text{otherwise.} \end{cases}$

For any $\mathbf{x}$, set C is unique and fixed. Therefore, AP can be translated to the problem of choosing B clusters that maximize total area burned:

$$\max_{\mathbf{y}} \quad \textstyle\sum_{c \in C} s_c \cdot y_c \tag{18}$$

$$\textstyle\sum_{c \in C} y_c \leq B \tag{19}$$

$$y_c \in \{0,1\} \qquad \forall c \in C \tag{20}$$

Problem (18)-(20) is a knapsack problem with unitary weights and can be solved to optimality by letting $y_c = 1$ for the B largest subclusters.

Following these considerations, a single-level reformulation of the problem that explicitly considers all the possible clusters resulting from the action of the defender is presented.

3.2. Single-Level Reformulation

The problem can be reformulated as a single-level integer programming (IP) problem by enumerating all the subclusters resulting from a feasible treatment. A treatment is feasible if it does not exceed the defender budget. A subcluster is a connected subset of the nodes in a cluster. Figure 3 illustrates an example.

In this problem, the defender chooses for each original cluster one of the feasible treatments, without exceeding the total treatment budget available. The treatments partition the clusters into subclusters. The attacker chooses the B largest subclusters among those resulting from the actions of the defender. The objective of the defender is to minimize the total area of the sub-clusters chosen by the attacker. The single-level formulation is presented in the following. Please note that the following notation supersedes any previous definition.

Sets

- C, set of clusters, indexed by i and j.
- T_i, set of feasible treatments for cluster i, including the option of not treating the cluster, indexed by k and l.
- C_i^k, set of subclusters that are obtained from cluster i by applying a feasible treatment $k \in T_i$, indexed by c and d.

Parameters

- $S > 0$, defender's treatment budget.
- $B \in \mathbb{N}$, number of subclusters chosen by the attacker (i.e., the attacker's capacity), indexed by $a, b = 1, \ldots, B$.
- $t_{ik} > 0$, cost of treatment $k \in T_i$.
- $s_{ikc} > 0$, area of subcluster $c \in C_i^k$.

Variables

- $x_{ik} = \begin{cases} 1 & \text{if the defender chooses to apply treatment } k \in T_i \text{ to cluster } i, \\ 0 & \text{otherwise.} \end{cases}$
- $y_{ikcb} = \begin{cases} 1 & \text{if the subcluster } c \in C_i^k \text{ is the } b\text{-th choice of the attacker given application of treatment } i, \\ 0 & \text{otherwise.} \end{cases}$

Formulation

Single-Level Problem [SLP]:

$$\min_{\mathbf{x}} \sum_{i\in C}\sum_{k\in T_i}\sum_{c\in C_i^k}\sum_{b=1}^{B} s_{ikc}\cdot y_{ikcb} \tag{21}$$

$$\text{s.t.} \sum_{k\in T_i} x_{ik} = 1 \qquad \forall i \in C \tag{22}$$

$$\sum_{i\in C}\sum_{k\in T_i} t_{ik}\cdot x_{ik} \le S \tag{23}$$

$$\sum_{b=1}^{B} y_{ikcb} \le x_{ik} \qquad \forall i \in C, k \in T_i, c \in C_i^k \tag{24}$$

$$\sum_{i\in C}\sum_{k\in T_i}\sum_{c\in C_i^k} y_{ikcb} = 1 \qquad \forall b = 1,\ldots,B \tag{25}$$

$$\sum_{i\in C}\sum_{k\in T_i}\sum_{c\in C_i^k} s_{ikc}\cdot y_{ikcb} \ge \begin{cases} s_{jld}\cdot x_{jl}, & \text{if } b=1. \\ s_{jld}\cdot (x_{jl} - \sum_{a=1}^{b-1} y_{jlda}), & \text{otherwise.} \end{cases} \quad \forall b = 1,\ldots,B, j\in C, l \in T_j, d \in C_j^l \tag{26}$$

$$x_{ik} \in \{0,1\} \qquad \forall i \in C, k \in T_i \tag{27}$$

$$y_{ikcb} \in \{0,1\} \qquad \forall i \in C, k \in T_i, c \in C_i^k, b = 1,\ldots,B \tag{28}$$

The objective is to minimise the total area of the subclusters chosen by the attacker (21). Constraints (22) and (23) concern the defender. The former states that the defender must choose exactly one treatment for every cluster, while the latter enforces the treatment budget. Constraints (24)-(26) are related to the attacker. The first set of constraints (24) limits the attacker's choices to the subclusters generated by the treatments carried out by the defender, while enforcing at the same time that each subcluster can be selected only once. The assignment constraints (25) obligate the attacker to choose exactly one subcluster for each unit of capacity. Constraints (26) state that the subclusters chosen by the attacker must be larger in terms of area than all the available subclusters that have not been chosen, effectively forcing the attacker to choose the B largest subclusters available in decreasing order of area. Finally, all the variables are required to be binary (27)-(28).

Model Analysis

A number of considerations on the formulation are presented.

Constraints (24) and (25) are redundant. In fact, constraints (26) implicitly enforce the same conditions. However, preliminary experiments show that their inclusion improves the solution process.

The current formulation might result in infeasible solutions when B is large. In fact, due to constraints (25), the defender must accommodate the requirement of the attacker to strike exactly B subclusters. When this is not possible, the problem is infeasible. Obviously, this behaviour of the defender is not desirable. Fortunately, this can be easily fixed by adding $(B - |C|)$ dummy clusters to C, when $B > |C|$. A dummy cluster is an empty cluster with only one treatment having cost and area equal to zero. In this way, the attacker can strike a dummy cluster if there are not enough subclusters, without affecting the objective function value.

Finally, under certain circumstances, the integrality condition on the $\mathbf{y}$ variables can be relaxed and allowed to be continuous in the range $[0, 1]$. However, these conditions cannot be assumed to be true for every problem instance. For more information, the reader is referred to the Appendix.

4. Implementation

The following section provides technical details of the methodology. Readers only interested in the practical outcomes of this work may omit this section.

SLP relies on the complete enumeration of all the feasible treatments that can be carried out by the defender on each cluster. This could lead to a combinatorial explosion that depends on many factors, including the number and size of hazardous clusters in the graph. Therefore, it is fundamental to apply smart ways of considering the smallest possible number of treatments in order to reduce as much as possible the size of the model, without losing optimality. In the following, the required concepts and implementation details are formally introduced.

4.1. Feasible Treatments Generation Procedure

For a given a cluster, all feasible treatments are generated by a traversal algorithm that explores a search tree. Every node of the tree represents a treatment and its root node is the empty treatment (i.e., the defender does not treat any node in the cluster). Feasible treatments whose cost of treatment is less than or equal to the defender's budget are generated by iteratively adding one element of the cluster to the parent treatment to form a child treatment. To avoid symmetries in the search tree, the element added to the parent treatment to generate a child must have an index greater than the last element of the parent treatment. Infeasible treatments are discarded from the tree. A sample search tree is displayed in Figure 4.

All the nodes in the tree correspond to feasible treatments. However, not all of them are required in the optimal solution. Therefore, a number of procedures have been implemented to reduce the set of treatments considered and improve the performance of the solution approach.

4.2. Dominance Rule for Treatments

Given a cluster i in C, a treatment $k \in T_i$ is characterised by its cost and the areas of the subclusters it generates, $(t_{ik}, \mathbf{s}_{ik})$, where $\mathbf{s}_{ik}$ is the vector of the subcluster areas, $\mathbf{s}_{ik} = (s_{ik1}, s_{ik2}, \ldots, s_{ik|C_i^k|})$. It is assumed that $\mathbf{s}_{ik}$ is sorted in non-increasing order. It is possible to establish a dominance rule among treatments of the same cluster in such a way that only efficient (i.e., non-dominated) treatments need to be included in the optimisation model.

Let $\bar{\mathbf{s}}_{\mathbf{ik}}$ be the vector of the cumulative sums of the subcluster areas vector:

$$\bar{\mathbf{s}}_{\mathbf{ik}} = \left(s_{ik1},\ s_{ik1} + s_{ik2},\ \ldots,\ \sum_{c \in C_i^k} s_{ikc} \right) \tag{29}$$

then a treatment k dominates a treatment l, i.e., $k \succ l,\ k, l \in T_i$, when:

$$\begin{gathered} t_{ik} \leq t_{il} \\ \bar{s}_{ik1} \leq \bar{s}_{il1} \\ \bar{s}_{ik2} \leq \bar{s}_{il2} \\ \cdots \\ \bar{s}_{ik|C_i^k|} \leq \bar{s}_{il|C_i^l|} \end{gathered} \tag{30}$$

and at least one inequality is strict. In the case that two treatments have different number of subclusters, then the shorter vector $\mathbf{s}_{ik}$ can be extended with zeros, and the vector of cumulative sums $\bar{\mathbf{s}}_{\mathbf{ik}}$ can be calculated as specified.

The rationale behind this dominance rule is that the defender is intelligent and, therefore, they will never choose a treatment that is worse and has cost greater than or equal to another. The cumulative sums represent the contribution of the treatment to the objective function. In particular, the n-th element of the cumulative sums vector represents the increase in the objective function that occurs should the attacker choose the largest n subclusters resulting from the treatment. Since the attacker cannot choose more than B subclusters in total, it is not necessary to compare more than B elements of the cumulative sum vectors. Following this consideration, for a specific value for an attacker's capacity, B, the dominance rule can be reduced as follows:

$$\begin{gathered} t_{ik} \leq t_{lk} \\ \bar{s}_{ik1} \leq \bar{s}_{il1} \\ \bar{s}_{ik2} \leq \bar{s}_{il2} \\ \cdots \\ \bar{s}_{ikm} \leq \bar{s}_{ilm} \end{gathered} \tag{31}$$

where $m = \min\{B, \max\{\left|C_i^k\right|, \left|C_i^l\right|\}\}$. The efficient subset identified by this rule is necessarily smaller than or equal to that resulting from applying the original dominance rule (30).

4.3. Cost Threshold Filtering Procedure

It is possible to obtain a threshold value $\underline{t}_i$ on the treatment cost in the optimal solution for each cluster i, corresponding to the maximum budget that the defender can spend on cluster i without limiting the choice of treatment for the other clusters. In other words, the defender can spend as much as $\underline{t}_i$ on cluster i and they will still be able to choose any possible treatment for the remaining clusters. Equation (32) illustrates how the threshold can be calculated:

$$\underline{t}_i = \max\{0, S - \sum_{j \in C: j \neq i} \max_{l \in T_j}\{t_{jl}\}\} \tag{32}$$

Equation (32) states that for cluster i, the corresponding threshold $\underline{t}_i$ is the difference between the treatment budget and the sum of maximum treatment costs for all the other clusters.

This value can be used to reduce the set of treatments to include in the optimisation model. Given a cluster i, let us consider the set of treatments having a cost less than or equal to the threshold: $\underline{T}_i = \{k \in T_i : t_{ik} \leq \underline{t}_i\}$. The treatment cost should not be considered when comparing two treatments belonging to set $\underline{T}_i$, as it is not a constraining factor on the actions of the defender.

This allows us to relax the reduced dominance rule (31) by removing the treatment cost inequality when comparing treatments belonging to $\underline{T}_i$. More formally, given two treatments $k, l \in T_i$ such that $t_{ik}, t_{il} \leq \underline{t}_i$, then k dominates a treatment l, i.e., $k \succ l$, when:

$$\begin{gathered} \bar{s}_{ik1} \leq \bar{s}_{il1} \\ \bar{s}_{ik2} \leq \bar{s}_{il2} \\ \cdots \\ \bar{s}_{ikm} \leq \bar{s}_{ilm} \end{gathered} \tag{33}$$

and at least one inequality is strict. Due to one fewer inequality, this relaxed dominance rule results in an efficient treatment set that is smaller than or equal to that of the reduced dominance rule (Equation 31).

4.4. Primal Solution Procedure for the SLP

A simple procedure to find a primal solution to SLP is now presented. In summary, the procedure starts from an initial trivial solution and improves it iteratively by identifying the largest subcluster and considering each treatment in non-increasing order of cost until the solution cannot be improved without exceeding the budget. The complete algorithm is illustrated in detail in Algorithm 1.

The algorithm assumes that the treatments in the sets T_i are sorted in non-decreasing order of cost. The initial solution is generated by assigning to each cluster the cheapest treatment (line 1). $\mathbf{K}$, $\overline{\mathbf{K}}$, and $\widehat{\mathbf{K}}$ are vectors of treatment indices that represent the current solution, the best solution found so far, and the next treatments that should be considered (line 2), respectively. Vector

Algorithm 1 Heuristic procedure for the solution of the SLP.

Input: $C,\ T_i,\ C_i^k,\ S,\ B,\ t_{ik},\ s_{ikc}$.
Output: $\overline{\mathbf{K}}$.

1: $\underline{\mathbf{K}} \leftarrow \overline{\mathbf{K}} \leftarrow \mathbf{1}$ ▷ Initial solution
2: $\widehat{\mathbf{K}} \leftarrow \mathbf{2}$ ▷ Next treatments to consider
3: **search** $\leftarrow (|T_i| > 1, \forall i \in C)$ ▷ Loop condition
4: $\hat{\imath} \leftarrow \arg\max_{i \in C: search_i} \{\max_{c \in C_i^{K_i}} \{s_{iK_ic}\}\}$ ▷ Current cluster with largest subcluster
5: **while** $(\exists i \in C : search_i)$ **do** ▷ Loop while some cluster can be improved
6: $\quad$ *updated* $\leftarrow$ false
7: $\quad K_{\hat{\imath}} \leftarrow \widehat{K}_{\hat{\imath}}$ ▷ Current solution update
8: $\quad$ **if** $(\sum_{i \in C} t_{iK_i} \le B)$ **then** ▷ Check solution feasibility
9: $\quad\quad$ **if** (**SolutionValue**($\mathbf{K}$) < **SolutionValue**($\overline{\mathbf{K}}$)) **then** ▷ Check improvement
10: $\quad\quad\quad \overline{\mathbf{K}} \leftarrow \mathbf{K}$ ▷ Update best solution
11: $\quad\quad\quad$ *updated* $\leftarrow$ true
12: $\quad\quad$ **end if**
13: $\quad\quad \widehat{K}_{\hat{\imath}} \leftarrow \widehat{K}_{\hat{\imath}} + 1$ ▷ Consider next treatment
14: $\quad\quad$ **if** $(\widehat{K}_{\hat{\imath}} > \left|T_{\hat{\imath}}\right|)$ **then** ▷ Check treatment availability
15: $\quad\quad\quad search_{\hat{\imath}} \leftarrow$ false
16: $\quad\quad$ **end if**
17: $\quad$ **else**
18: $\quad\quad search_{\hat{\imath}} \leftarrow$ false ▷ Current cluster cannot be improved
19: $\quad$ **end if**
20: $\quad$ **if** $((\exists i \in C : search_i) \wedge (updated \vee \neg search_{\hat{\imath}}))$ **then** ▷ Check need to change current cluster
21: $\quad\quad \hat{\imath} \leftarrow \arg\max_{i \in C: search_i} \{\max_{c \in C_i^{K_i}} \{s_{iK_ic}\}\}$ ▷ Current cluster with largest subcluster
22: $\quad$ **end if**
23: **end while**
24: **return** $\overline{\mathbf{K}}$ ▷ Return best solution found

search is a vector of logical values whose elements state if the corresponding cluster can be improved (line 3). The improvable cluster containing the largest subcluster is identified and stored in $\hat{\imath}$ (line 4). The main body of the algorithm follows. The procedure iteratively looks for a cluster that can be improved (line 5). This is done by considering the next treatment for the current cluster (line 7). If the resulting solution is feasible (line 8), then the algorithm checks if it is better than the current best solution (line 9) and, if that is the case, then the former replaces the latter (line 10). Next, vector $\widehat{\mathbf{K}}$ is updated to consider the next treatment for the current cluster (line 13) and the cluster is excluded from the search if all of its treatments have been considered (lines 14 and 15). If, on the other hand, the current solution is not feasible, then the current cluster cannot be further improved (line 18), as any remaining treatment is too expensive. The last part of the loop updates the cluster considered if there are clusters that can be improved and the best solution has been updated or the current cluster cannot be improved anymore (line 20). In that case, the improvable cluster having the largest subcluster is identified (line 21). Finally, the procedure returns the best solution found (line 24).

4.5. Primal Bound Based Filtering Procedure

The value provided by the primal solution procedure can be used to remove treatments that necessarily cannot be part of the optimal solution.

Suppose that a primal (upper) bound $\overline{Z}$ to the optimal value $Z^\star$ is given (i.e., $\overline{Z} \geq Z^\star$). Let us consider a cluster i and a treatment k. $\underline{Z}_{ik}$ is a dual bound to the solution value obtained after choosing treatment k for a cluster i in the solution. We can remove treatment k from the model if:

$$\underline{Z}_{ik} > \overline{Z} \ (\geq Z^\star) \tag{34}$$

as treatment k can only be included in sub-optimal solutions.

A trivial dual bound $\underline{Z}_{ik}$ is given by considering that all clusters except i are fully treated, that is, they have no subclusters. This is equivalent to relaxing constraint (23) in SLP. In this case, the only cluster that can have subclusters and that affects the solution value is the incumbent one (i), and the dual solution value is equal to the sum of its subclusters' areas. Therefore, the filtering rule becomes:

$$\underline{Z}_{ik} = \bar{s}_{ikm} > \overline{Z} \ (\geq Z^\star) \tag{35}$$

In conclusion, a treatment can be excluded from the model if the sum of the subclusters it generates is larger than the upper bound.

4.6. Proposed Algorithm for the SLP

The algorithm implemented applies the methodologies presented above in a specific order, according to their complexity.

- **Step 1:** Feasible treatments generation procedure.
- **Step 2:** Cost threshold filtering procedure.
- **Step 3:** Primal solution procedure.
- **Step 4:** Primal bound based filtering procedure.
- **Step 5:** Filtering based on the reduced dominance rule.
- **Step 6:** Solution of the SLP with the treatment set resulting from the previous steps.

Firstly, all the feasible treatments are generated by the *feasible treatments generation procedure*. Then, the *cost threshold filtering procedure* is applied. The threshold values are very quick to compute. In the worst case, applying filtering rule involves comparing all the pairs of elements, thus, resulting in quadratic complexity. However, in practice, most of the treatments are dominated by the most expensive one as, in general, a higher treatment cost should lead to smaller subclusters (note that this is not always true). Therefore, the procedure is applied starting from the most expensive treatments in each set $\underline{T}_i$, resulting in almost linear complexity. Next, the treatments that pass this filter are given as inputs to the *primal solution procedure*. The complexity of this procedure is linear in the number of treatments. However, calculating the value of a solution involves sorting the array of subcluster sizes. Therefore, the total complexity is greater than linear. The fourth step is the *primal bound based filtering procedure*, which has linear complexity. Then, *filtering based on the reduced dominance rule* is applied, which has quadratic complexity and is the most expensive step. This time, the procedure starts applying the reduced dominance rules from the less expensive treatments. Finally, the treatment set resulting from the filtering procedures is included in SLP, which is solved by a commercial solver.

5. Computational Experiments

The solution algorithm was programmed in Julia v.1.4.0 (Bezanson et al., 2017). SLP has been implemented in JuMP v.0.21.2 (Dunning et al., 2017) and solved using Gurobi v.9.0.1. (Gurobi Optimization, 2020). All experiments were run on a Dell Precision 5540 equipped with 16-core Intel i9-9880H CPU (2.30GHz per core) and 16GB RAM. The standard configuration of Gurobi was used, which applies multithreading. A CPU time limit of 3600s and a virtual memory limit of 16GB was set for all optimisation runs.

For the experiments, problem instances similar to the one illustrated in Figure 1 were randomly generated, as explained in the following. A number of points (*nodes*) were randomly distributed (uniformly) on a square plane having an area of 100. Each point represents a node in the graph and some of them are marked as hazardous according to a certain probability (p_f). The Voronoi diagram induced by the points was used to obtain each nodes' area and neighbours. Two nodes are connected by an edge in the graph if their corresponding cells share a side.

To test the algorithm, five random instances were generated for each combination of the following parameters:

- $nodes = \{25,\ 50,\ 100\}$
- $p_f = \{0.3,\ 0.5,\ 0.7\}$

Each instance was solved once for every combination of the following model parameters:

- $S = \{2.5,\ 5,\ 10,\ 20\}$.
- $B = \{1,\ 2,\ 3,\ 4\}$.
- $filters = \{true,\ false\}$.

Where S is the defender's treatment budget, B the number of subclusters chosen by the attacker, and the parameter *filters* makes reference to the use of the filtering procedures applied to the treatments (i.e., steps 2–5 in the proposed algorithm for SLP).

Overall, 45 instances and 32 model configurations were considered, corresponding to 1,440 problems solved. Due to its large size, a table showing individual problem results is not reported in the paper. However, the interested reader can download the problem instances and the results table from Liberatore et al. (2020).

The remainder of this section concerns the data analysis carried out to draw insights on the solution algorithm and its components. In particular, the analysis focuses on the number of instances solved, solution time, and the quality of the primal solution found by the primal bound procedure. All statistical tests use a significance level of $\alpha = 0.05$.

5.1. Number of Instances Solved

In the following tests, observations are grouped according to the parameter *filters* to analyse the effect of the filters on the number of instances solved within the limits. Table 1 shows a summary.

Table 1: Number of problem instances solved, not solved, and total number for each group.

filters	solved	not solved	total
true	605	115	720
false	460	260	720

The filters allow 605/720 problem instances to be solved, corresponding to 44% more problem instances solved than the group without filters. This difference is statistically significant: $\mathrm{p-value} < 2.2 \times 10^{-16}$ in a test for equality of proportions.

5.2. Solution Time

The solution time includes the time necessary to create and filter the treatments and to build and solve the SLP optimisation model. Summary statistics are presented in Table 2. These statistics consider only the problems that have been solved to optimality within the time and memory limits (i.e., 3600s and 16GB, respectively). The second row ($true^{\star}$) refers to the group *filters* = *true* considering only the subset instances that could be solved to optimality by the group *filters* = *false*.

Table 2: Solution time summary statistics (in seconds) for each group.

filters	Min.	Q1	Median	Mean	Q3	Max.	St. Dev.
true	0.003	0.005	0.014	16.363	0.063	2651.379	148.9022
$true^{\star}$	0.003	0.005	0.007	11.8927	0.0392	2651.3790	133.0131
false	0.003	0.010	0.050	57.596	2.508	3337.934	285.8658

On average, the proposed algorithm took approximately 16 seconds to solve each problem instance. However, as it can be deduced from the quantiles and the large difference between the mean and the median, the solution time distribution is strongly right-skewed. Therefore, the average solution time is not representative. On the other hand, the table allows to evaluate the impact of the filters on solution time. By comparing the second and third rows, it can be seen that, on average, using the filters reduces the computational time by 79.35%. This difference is statistically significant: $\mathrm{p-value} < 2.2 \times 10^{-16}$ using a Wilcoxon signed rank paired test. The non-parametric Shapiro-Wilk normality test reveals that the differences in solution times between the two groups are not normal: $\mathrm{p-value} < 2.2 \times 10^{-16}$.

5.3. Primal Bound Analysis

The following subsection makes reference to technical details of the methodology. Readers only interested in the practical outcomes of this work may omit this section.

The quality of the primal solution found by the primal bound procedure presented in Section 4 is assessed by measuring its gap to the optimal solution:

$$\overline{gap} = \frac{\overline{Z} - Z^{\star}}{\overline{Z}} \tag{36}$$

The gap is calculated only for the problems that have been solved to optimality within the limits. Summary statistics are presented in Table 3. The last column shows the number of problems where the gap between the primal bound and the optimal solution is zero over the total number of problems solved to optimality.

Table 3: Summary statistics for $\overline{gap}$.

Min.	Q1	Median	Mean	Q3	Max.	#zeros/total
0.00000	0.00000	0.00000	0.00548	0.00000	0.18160	523/605

In more than 86% of the problem instances the primal bound procedure was able to identify an optimal solution. For the remaining instances the gap is still very low. To have a better understanding of the distribution of $\overline{gap}$, the quantiles 85% to 100% (with 1% increments) are given in Table 4.

Table 4: $\overline{gap}$ quantiles from 85% to 100% in 1% increments.

quantile	85%	86%	87%	88%	89%	90%	91%	92%
value	0.00000	0.00000	0.00013	0.00237	0.0079	0.01223	0.01696	0.01872

quantile	93%	94%	95%	96%	97%	98%	99%	100%
value	0.02316	0.02547	0.03815	0.05056	0.0634	0.07994	0.11144	0.18160

Overall, the upper bound procedure performs extremely well. It has achieved a $\overline{gap} < 0.01$ in 89% of the problems considered, and a $\overline{gap} < 0.05$ in 95% of them.

6. Case Study

The methodology proposed was tested on a real-world case study (León, 2020) concerning a territory in Andalusia, a region in southern Spain, located across two mountain ranges: *Sierra de Baza* (Granada) and *Sierra de Los Filabres* (Almería). The area considered, displayed in Figure 5, has an extension of 1820km^2. It includes the Sierra de Baza National Park and is under the jurisdiction of the Group for the Prevention and Extinction of Forest Fires of Andalusia (INFOCA).

The landscape was provided by INFOCA that identified the burn units according to the territory's topology, vegetation, and land ownership. It is comprised of 193 burn units, 33 of which are hazardous. Eighty units are private, which implies that they cannot be treated by the defender but they can still be affected by the attacker and by fire. This has been implemented in SLP by making a minor adjustment to the *feasible treatments generation procedure*: only the nodes corresponding to public burn units are considered when generating the children treatments in the procedure; however, all the hazardous nodes are included in the computation of the subclusters' areas. The dataset of the case study can be downloaded from Liberatore et al. (2020).

Regarding the model parameters, the value of the defenders' budget was determined considering the area treated every year by INFOCA and it has been set to $S = 150(\text{km}^2)$, while the attacker's capacity (i.e., the number of simultaneous ignition points) was set to $B = 1, \ldots, 4$ to examine for different scenarios. The total number of feasible treatments for the case study with the considered defender's budget is 1,434,965. Table 5 shows statistics on the optimisation model and the solutions.

Table 5: Model and solution statistics.

B	*treatments*	*time* (s)	$Z^\star$	$\overline{Z}$	$\overline{gap}$
1	4	18.81	64.32	64.32	0
2	4	17.96	124.59	124.59	0
3	4	18.365	181.23	181.23	0
4	5	18.486	233.19	233.19	0

The columns in the table correspond to the following information: the attacker's capacity (B), the number of treatments included in the SLP after the filtering procedures (*treatments*), the total solution time (*time*), the optimal objective ($Z^\star$, i.e., the total surface affected by the wildfire), the upper bound ($\overline{Z}$, i.e., the value of the solution identified by Algorithm 1), and the corresponding gap ($\overline{gap}$). From the table, it can be concluded that the algorithm proposed is very effective at solving a real-world problem instance, taking less than 19 seconds for a problem that needs to be solved once a year. Also, the treatments filtering procedures are extremely effective. In fact, the total number of feasible treatments for the case study is 1,434,965 and, as shown in the table, only 4 or 5 treatments are included in the SLP. Finally, the primal bound procedure always identifies the optimal solution.

The solutions are represented in Figure 6. The figures show that for this specific case study, the attacker's capacity does not have a strong impact on the defender's strategy. In fact, the defender treats the same nodes for $B = 2, 3, 4$, and for $B = 1$ the strategy differs only by one node. This is actually good news for INFOCA, as the solution for $B = 2$ is highly resilient to changes to the number of ignition points.

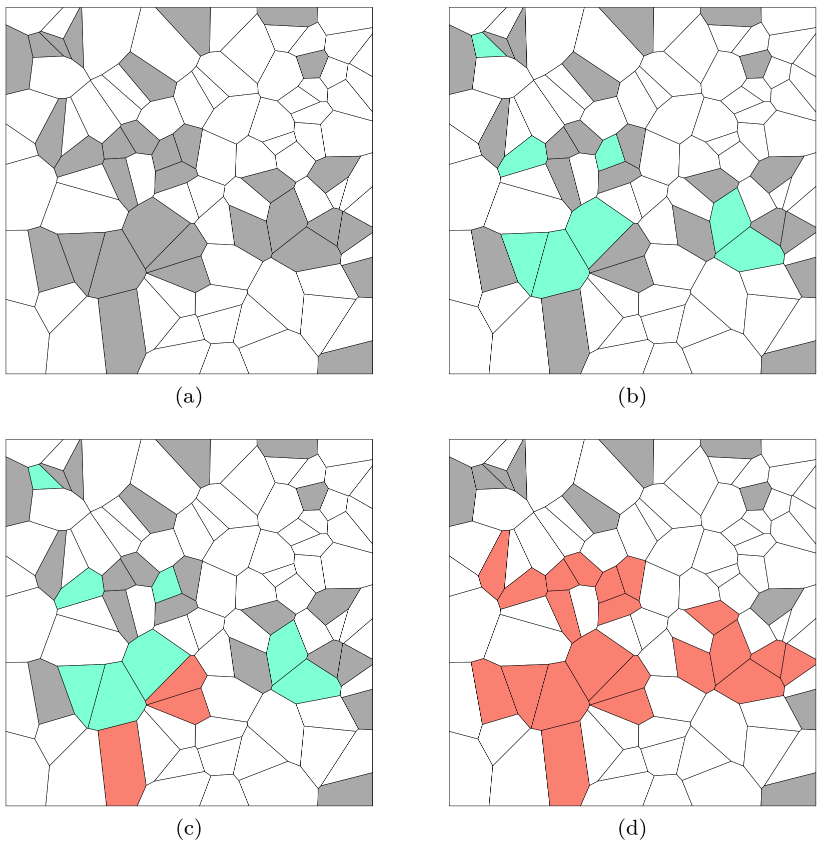


Figure 1: A landscape is presented in (a). The landscape, whose total area is 100, is divided into units. The hazardous units are colored in grey. The defender strategy is shown in (b). Assuming that the defender's budget is 10, the units in green are treated and, therefore, they are non-hazardous. Finally, the attack (i.e., the worst-case wildfire) is illustrated in (c), assuming two ignition points. The cells in red are burned by the wildfire (burned area: 4.25). If the defender would not have applied the treatment, the result would have been the wildfire portrayed in (d) (burned area: 21.92).

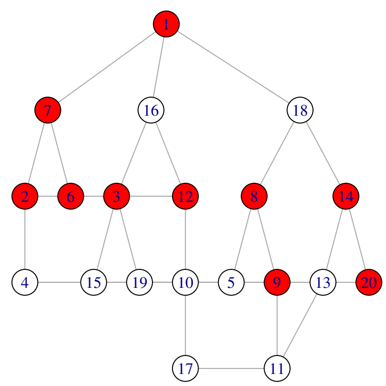


Figure 2: Example of clusters in a graph. The graph is comprised of 20 nodes, each identified by a number. The edges between nodes are represented by lines. The nodes in red are hazardous while the nodes in white are not. This graph presents three clusters: $\{1, 7, 2, 6, 3, 12\}$, $\{8, 9\}$, and $\{14, 20\}$.

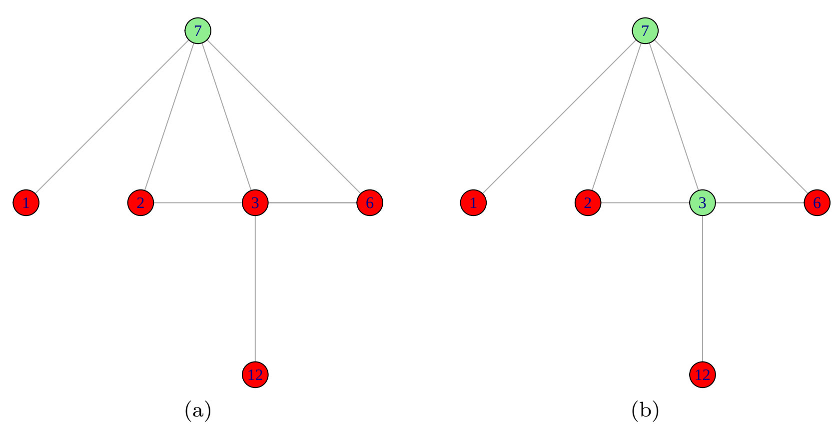


Figure 3: Subcluster examples. Cluster $\{1, 7, 2, 6, 3, 12\}$ from Figure 2 is considered. Treated nodes are colored in green. On the left (a), node 7 is treated, generating two subclusters: $\{1\}$ and $\{2, 3, 6, 12\}$. On the right (b), nodes 7 and 3 are treated, generating four subclusters: $\{1\}$, $\{2\}$, $\{6\}$, and $\{12\}$.

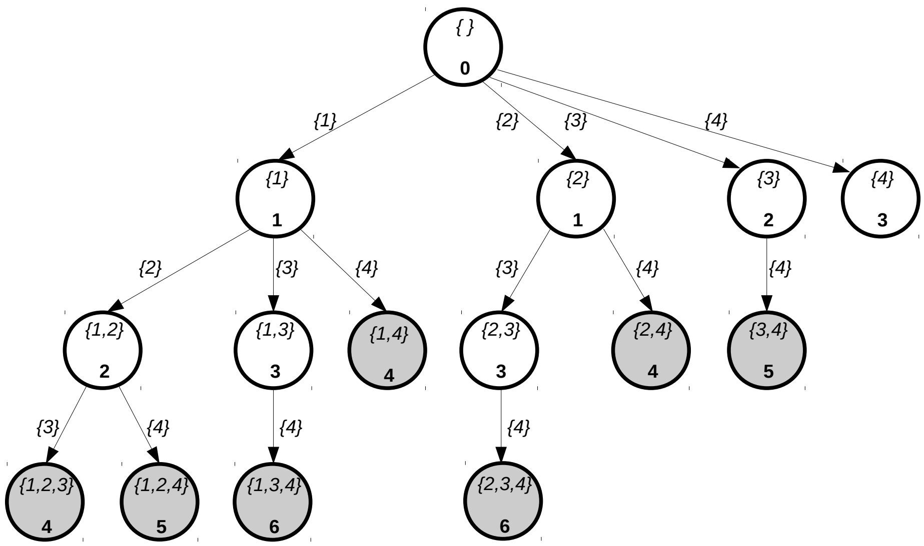


Figure 4: Example of a complete search tree corresponding to a fully connected cluster comprised of the nodes $N_i = \{1, 2, 3, 4\}$ having an area of $\mathbf{s}_n = \{1, 1, 2, 3\}$. The defender budget is $S = 3$. Each node shows the treatment set (top) and its cost (bottom). The arcs connect a parent treatment to its children treatments, which are generated by adding to the parent treatment the destination node of the corresponding arc. The nodes in grey are infeasible (i.e., their cost exceeds the defender budget). The considered cluster has eight feasible treatments.

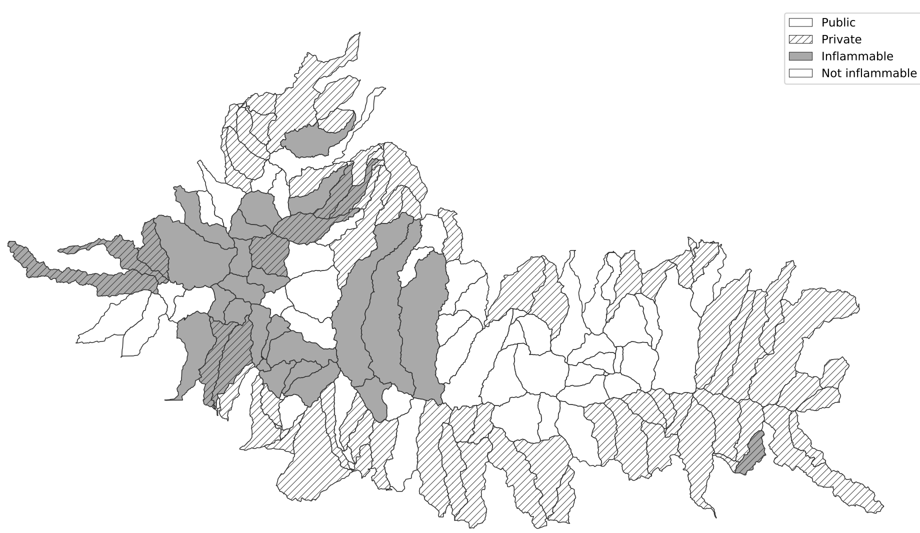
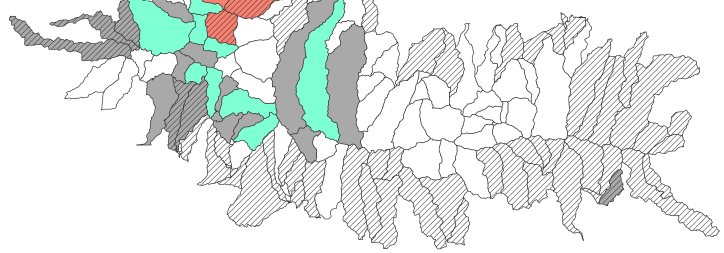
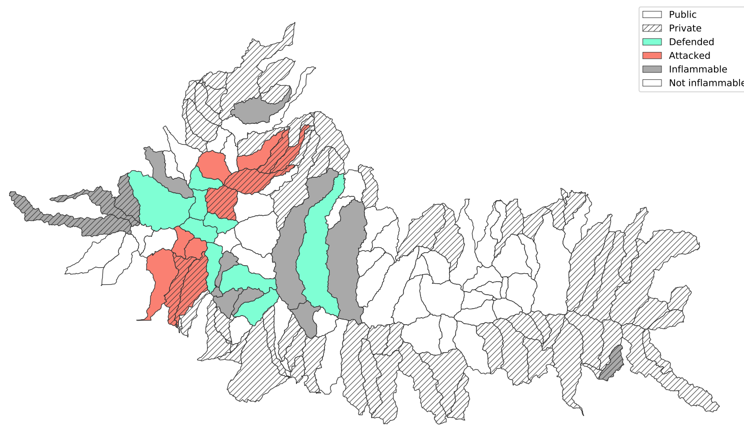


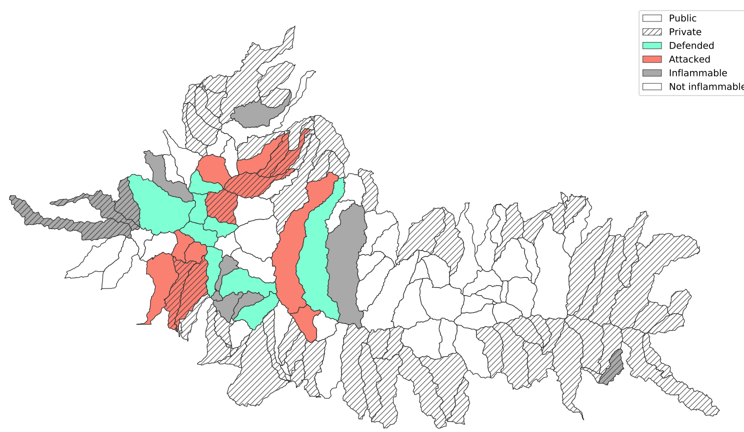
Figure 5: Case study landscape representation. The data presents three clusters: two single-unit clusters and one macro-cluster formed by 31 units.



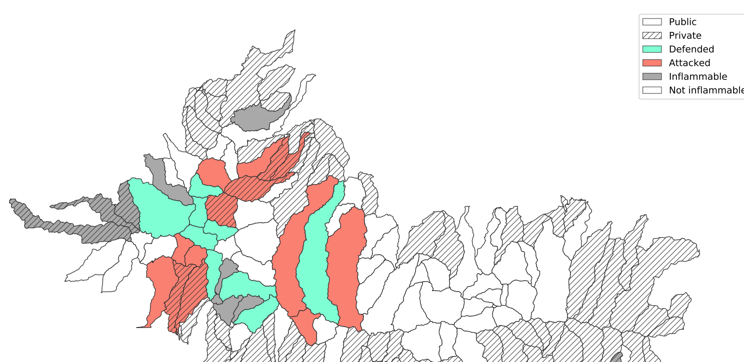
(a) $B = 1$


Public
Private
Defended
Attacked
Inflammable
Not inflammable

(b) $B = 2$


Public
Private
Defended
Attacked
Inflammable
Not inflammable

(c) $B = 3$


Public
Private
Defended
Attacked
Inflammable
Not inflammable

7. Suggestions for Implementation

The model presented in this paper requires a certain number of parameters to be run. In particular, it relies on representing the landscape as a graph. Most of the parameters should be specified by the decision-maker or other experts with an in-depth knowledge of the territory. In the following, guidelines on how to define them are given.

7.1. Node Parameters

In the model, each node of the graph represents a burn unit. The specification of the burn units is left to the expert decision-maker, although it is recommended that each burn unit is internally homogeneous according to relevant characteristics, such as vegetation type, slope, and aspect.

Additionally, the decision-maker must decide which nodes should be labeled as "hazardous". It is important to notice that the model considers that only hazardous nodes can be affected by a wildfire, and disregards the rest. Normally, a decision-maker would determine the set of hazardous nodes by setting a threshold on certain attributes of the burn unit, such as fuel load or vegetation. A risk-averse decision-maker might decide to label as "non-hazardous" only units that have no vegetation at all. In this respect, the model could be used to identify treatment strategies at different risk thresholds, so to provide a complete spectrum of solutions and help the decision-maker to identify the best course of action.

7.2. Edge Parameters

Edges connect nodes and are used to specify potential travelling directions for the fire. It is important to notice that the edges in the model are directed. Therefore, they can be used to represent different problem characteristics, as explained in the following.

- **Contiguity**: two burn units are contiguous if they share part of their perimeter. To allow the fire to spread from one unit to the other, the corresponding nodes (i and j) should be connected by two directed edges, one going from i to j, and the other going from j to i.

- **Obstacles**: contiguous burn unit might be separated by obstacles which are insurmountable to the fire. In this case, the edges connecting the nodes should not be included in the graph.

- **Slopes**: slopes might force the fire to spread only in a specific direction. To add this behaviour to the model, two nodes corresponding to burn units at different height should be connected with a single edge, going from the lowest unit to the highest.

- **Wind**: similarly to slopes, the wind might force fire to spread only in a specific direction. To introduce this behaviour in the model, every pair of contiguous nodes should be connected by a single downwind edge.

– **Spot fires**: the behaviour of spot fires can be modelled by connecting with edges all the pairs of nodes that are within a certain distance.

8. Conclusions

The purpose of this work was to address the problem of planning for fuel treatments in a landscape to buffer wildfires. The problem was tackled by formulating an optimisation model that identifies effective treatment configurations according to budget restrictions. Extensive tests showed that the proposed solution approach is capable of solving realistic-size instances in a reasonable amount of time. Also, the methodology was successfully applied to a real-world case study on a landscape in Andalusia, Spain. The approach can be extended in several ways to give rise to more realistic, useful, applicable, and informative models:

– The current model focuses on the tactical phase of the problem by identifying treatment strategies for a single period, or stage. A multi-stage model would allow to strategically plan for several periods.

– The model could be adapted to consider environmental elements for the preservation of protected species.

– The introduction of different treatment goals in the model, such as, reducing fire severity, reducing fire spread rate, and facilitating suppression (Ager et al., 2013), would allow exploring their effects on the solutions obtained.

– Fire suppression decisions could be introduced in the model to jointly optimise fire management operations both before and during a wildfire.

More details are provided in the following subsections.

8.1. Summary of the Methodology

In this paper, a novel model for the optimisation of fuel management operations in a landscape to buffer wildfires is introduced. The model considers the underlying uncertainty by implementing a defender-attacker structure aimed at mitigating against the impact of the worst-case loss. The model, bilevel in nature, was reformulated as a single-level MIP and an efficient and effective algorithm was proposed to solve it to optimality. The algorithm relies on bounds to reduce the number of treatments that have to be included in the model without losing optimality.

8.2. Summary of the Results

The algorithm was tested on randomly generated instances using a wide range of parameters to assess its performance under a variety of application contexts. The results show that the filtering procedures devised allow 44% more problem instances to be solved and an average 79% reduction in solution time.

The primal bound computed during the solution procedure found an optimal solution 86% of the time and produces a very low gap in the remaining ones. The methodology was then applied to a real-world landscape in Andalusia, Spain. This case study illustrates that the proposed algorithm is extremely efficient and capable of solving a real-world problem in less than 20 seconds.

8.3. Future Research

These promising results open up a wide range of future lines of research. The model studied in this paper defines the fuel management operations for one year, therefore, it focuses on the tactical aspects of the problem. It would be interesting to extend it by considering a time horizon of several years, moving the focus from tactics to strategy. This would result in a multi-stage problem. Multi-stage two-players games are extremely complex. The classical approach to solving such problems is by using a search procedure such as the minimax algorithm. The minimax algorithm suffers from the curse of dimensionality and, therefore, some improvements have been proposed in the literature, such as alpha–beta pruning. Alpha-beta pruning relies on trivial bounds to disregard parts of the search trees which cannot lead to the optimal solution, similar to the branch and bound algorithm. Better bounds that should further improve the performance of the solution procedure can be obtained by solving multiple instances of the single-stage problem. Therefore, the optimal solution of the multi-stage problem hinges on efficiently solving the problem presented in this paper.

As a consequence, it is still desirable to further improve the performance of the solution algorithm proposed. Along these lines, more efficient treatment filters could be devised. In particular, dual bounds on the optimal solution value could potentially be used to filter out high-cost treatments. Preliminary experiments showed that the dual bound obtained from the linear relaxation of the SLP was not tight enough to result in a significant reduction of the treatments set. Thus, other types of bounds must be investigated.

An alternative would be to consider a different treatment generation algorithm than the one proposed. The current procedure relies on the explicit enumeration of all treatments that can be applied to a cluster, which are then filtered out. An approach that implicitly enumerates the treatments could be studied to limit the computational time and memory necessary to the generation of the model.

In terms of modelling, the problem could be expanded to consider environmental elements for the preservation of protected species. Interestingly, the inclusion of environmental constraints should have a positive impact on solution time for the multi-stage model, as they would allow excluding search paths that lead to solutions that do not satisfy environmental goals. Also, the problem could incorporate different treatment goals (i.e., reduce fire severity, reduce fire spread rate, facilitate suppression; see Ager et al. (2013)) which could result in competing objectives. The final model would be multi-criteria and multi-level in nature.

A different line of research could deal with the extension of the model to consider post-disaster fire suppression decisions. This could require the inclusion of additional information on the fire behaviour, possibly obtained by a wildfire spread model (Ager et al., 2014; Alcasena et al., 2019).

The authors hope that this work will be a useful source of ideas for future research on fuel management and contributes further to the development and solution of more complex and more realistic models for fire management.

Appendix A. On the Relaxation of the Attacker's Variables

Lemma 1. *In the SLP, the domain of variables* $\mathbf{y}$ *can be replaced to the unit interval if, for every feasible* $\mathbf{x}$*, there is only one subcluster having the largest size.*

Proof. Suppose the defender variables are fixed, and that $d_1 \in C_{j_1}^{l_1}$ is the largest subcluster present after the defender's treatments, with $x_{j_1 l_1} = 1$. By constraint (26):

$$\sum_{i \in C} \sum_{k \in T_i} \sum_{c \in C_i^k} s_{ikc} \cdot y_{ikcb} \geq \begin{cases} s_{jld} \cdot x_{jl}, & \text{if } b = 1. \\ s_{jld} \cdot (x_{jl} - \sum_{a=1}^{b-1} y_{jlda}), & \text{otherwise.} \end{cases} \quad \forall b = 1, \ldots, B, j \in C, l \in T_j, d \in C_j^l$$

Letting $b = 1, j = j_1, l = l_1, d = d_1$:

$$s_{j_1 l_1 d_1} \leq \sum_{i \in C} \sum_{k \in T_i} \sum_{c \in C_i^k} s_{ikc} \cdot y_{ikc1} \tag{A.1}$$

As y_{ikc1} can be greater than zero only on those existing subclusters (by constraint (24)) and their sum equals one (constraint (25)), the right-hand side is a convex combination of the sizes of the existing subclusters. Hence, given that $s_{j_1 l_1 d_1}$ is the size of the largest existing subcluster, for the inequality (A.1) to hold it is necessary that $y_{j_1 l_1 d_1 1} = 1$.

If $b = 2$, let $d_2 \in C_{j_2}^{l_2}$ be the second largest subcluster present after the defender's treatments. As $(j_1, l_1, d_1) \neq (j_2, l_2, d_2)$, $y_{j_2 l_2 d_2 1} = 0$. Consequently, an analogous argument leads to $y_{j_2 l_2 d_2 2} = 1$ and, by induction, to all b. □

When arriving to: $s_{j_1 l_1 d_1} \leq \sum_{ikc} s_{ikc} \cdot y_{ikc1}$, with $\sum_{ikc} y_{ikc1} = 1$, it is argued that $y_{ikc1} = 1$ on the largest available subcluster (j_1, l_1, d_1). However, if there are multiple subclusters attaining the maximum value, their associated variables y can be non-zero. This later causes that, for $b = 2$, there can be "some y" in $y_{j_2 l_2 d_2 2}$, so an analogous reasoning cannot be done, since $x_{j_2 l_2} - y_{j_2 l_2 d_2 2} \neq 1$.

Acknowledgements

The authors would like to thank INFOCA (Group for the Prevention and Extinction of Forest Fires of Andalusia) for all their help, support, and for providing the data for the case study.

Funding

The research is partially funded by the European Commission's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie, grant number MSCA-RISE 691161 (GEO-SAFE), and the Government of Spain, grant MTM2015-65803-R. The research of J. León has been partially funded by the UCM-Santander grant CT27/16-CT28/16. All the financial support is gratefully acknowledged.

Abbreviations

The following abbreviations are used in this manuscript:

MIP	Mixed integer programming
MTT	Minimum travel time
SP	Stochastic programming
RO	Robust optimisation
PTMP	Pyro-terrorism mitigation problem
VAIAP	Vulnerability assessment of the initial attack problem
DP	Defender problem
AP	Attacker problem
SM	System model
LBP	Linear bilevel program
KKT	Karush–Kuhn–Tucker
MIBP	Mixed integer bilevel programming
DAP	Dual attacker problem
LP	Linear programming
MILP	Mixed integer linear programming
IP	Integer programming
SLP	Single-level problem
GAM	Generalised additive model
INFOCA	Group for the Prevention and Extinction of Forest Fires of Andalusia

References

Adams, W.P., Forrester, R.J., 2005. A simple recipe for concise mixed 0-1 linearizations. Operations Research Letters 33, 55–61.

Ager, A.A., Day, M.A., Finney, M.A., Vance-Borland, K., Vaillant, N.M., 2014. Analyzing the transmission of wildfire exposure on a fire-prone landscape in oregon, usa. Forest Ecology and Management 334, 377–390.

Ager, A.A., Vaillant, N.M., McMahan, A., 2013. Restoration of fire in managed forests: a model to prioritize landscapes and analyze tradeoffs. Ecosphere 4, 1–19.

Alcasena, F.J., Ager, A.A., Bailey, J.D., Pineda, N., Vega-García, C., 2019. Towards a comprehensive wildfire management strategy for mediterranean areas:

framework development and implementation in catalonia, spain. Journal of environmental management 231, 303–320.

Bard, J.F., 2013. Practical bilevel optimization: algorithms and applications. volume 30. Springer Science & Business Media.

Bard, J.F., Falk, J.E., 1982. An explicit solution to the multi-level programming problem. Computers & Operations Research 9, 77–100.

BBC News, 2020. Australia fires: A visual guide to the bushfire crisis. URL: https://www.bbc.com/news/world-australia-50951043. accessed February 15, 2021.

Ben-Tal, A., Nemirovski, A., 2002. Robust optimization–methodology and applications. Mathematical Programming 92, 453–480.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM review 59, 65–98.

Brown, G.G., Carlyle, W.M., Salmeron, J., Wood, K., 2005. Analyzing the vulnerability of critical infrastructure to attack and planning defenses, in: Emerging Theory, Methods, and Applications. Informs, pp. 102–123.

Camacho-Collados, M., Liberatore, F., 2015. A decision support system for predictive police patrolling. Decision Support Systems 75, 25–37.

Chang, C.T., 2000. An efficient linearization approach for mixed-integer problems. European Journal of Operational Research 123, 652–659.

Church, R.L., Scaparra, M.P., 2007. Protecting critical assets: the r-interdiction median problem with fortification. Geographical Analysis 39, 129–146.

Dunning, I., Huchette, J., Lubin, M., 2017. Jump: A modeling language for mathematical optimization. SIAM Review 59, 295–320.

Fernandes, P., 2013. Forest fuel management for fire mitigation under climate change. Nova Science Publishers, Inc.. chapter 3. pp. 31–41.

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

Gillen, C.P., Matsypura, D., Prokopyev, O.A., 2017. Operations research techniques in wildfire fuel management, in: Optimization Methods and Applications. Springer International Publishing, pp. 119–135.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Management Science 22, 455–460.

Gorissen, B.L., Yanıkoğlu, İ., den Hertog, D., 2015. A practical guide to robust optimization. Omega 53, 124–137.

Gurobi Optimization, L., 2020. Gurobi optimizer reference manual. URL: http://www.gurobi.com. accessed February 15, 2021.

Husari, S., Nichols, H.T., Sugihara, N.G., Stephens, S.L., 2006. Fire and Fuel Management. 1 ed.. University of California Press. chapter 19. pp. 444–465.

Kabli, M., Gan, J., Ntaimo, L., 2015. A stochastic programming model for fuel treatment management. Forests 6, 2148–2162.

Kettani, O., Oral, M., 1990. Equivalent formulations of nonlinear integer problems for efficient optimization. Management Science 36, 115–119.

Lee, H., 2020. Bushfires Release Over Half Australia's Annual Carbon Emissions. Time Magazine URL: https://time.com/5754990/australia-carbon-emissions-fires/. accessed February 15, 2021.

León, J., Reijnders, V.M.J.J., Hearne, J.W., Ozlen, M., Reinke, K.J., 2019. A landscape-scale optimisation model to break the hazardous fuel continuum while maintaining habitat quality. Environmental Modeling & Assessment 24, 369–379.

León, J., 2020. Mathematical Programming with Uncertainty and Multiple Objectives for Sustainable Development and Wildfire Management. *Programación Matemática con Incertidumbre y Múltiples Objetivos para Desarrollo Sostenible y Gestión de Incendios Forestales*. Ph.D. thesis. RMIT University, *Universidad Complutense de Madrid*. URL: https://eprints.ucm.es/59548/. accessed February 15, 2021.

Liberatore, F., 2012. Protection planning for critical infrastructure systems in location analysis: models and algorithms. 4OR 10, 315.

Liberatore, F., León, J., Hearne, J., Vitoriano, B., 2020. Dataset and results repository. URL: http://blogs.mat.ucm.es/humlog/fuelmanagement/. accessed February 15, 2021.

Liberatore, F., Scaparra, M.P., 2011. Optimizing protection strategies for supply chains: comparing classic decision-making criteria in an uncertain environment. Annals of the Association of American Geographers 101, 1241–1258.

Liberatore, F., Scaparra, M.P., Daskin, M.S., 2011. Analysis of facility protection strategies against an uncertain number of attacks: The stochastic r-interdiction median problem with fortification. Computers & Operations Research 38, 357–366.

Liberatore, F., Scaparra, M.P., Daskin, M.S., 2012. Hedging against disruptions with ripple effects in location analysis. Omega 40, 21–30.

Marino, E., Hernando, C., Planelles, R., Madrigal, J., Guijarro, M., Sebastian, A., 2014. Forest fuel management for wildfire prevention in spain: a quantitative swot analysis. International Journal of Wildland Fire 23, 373–383.

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

Matousek, J., Gärtner, B., 2007. Understanding and using linear programming. Springer Science & Business Media.

Matsypura, D., Prokopyev, O.A., Zahar, A., 2018. Wildfire fuel management: network-based models and optimization of prescribed burning. European Journal of Operational Research 264, 774–796.

Minas, J., Hearne, J., Martell, D., 2015. An integrated optimization model for fuel management and fire suppression preparedness planning. Annals of Operations Research 232, 201–215.

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, 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, 412–422.

Olson, J.S., 1963. Energy storage and the balance of producers and decomposers in ecological systems. Ecology 44, 322–331.

Pacheco, A.P., Claro, J., Fernandes, P.M., de Neufville, R., Oliveira, T.M., Borges, J.G., Rodrigues, J.C., 2015. Cohesive fire management within an uncertain environment: a review of risk handling and decision support systems. Forest Ecology and Management 347, 1–17.

Qiao, J., Jeong, D., Lawley, M., Richard, J.P.P., Abraham, D.M., Yih, Y., 2007. Allocating security resources to a water supply network. IIE Transactions 39, 95–109.

Rachmawati, R., Ozlen, M., Hearne, J., Reinke, K., 2018. Fuel treatment planning: Fragmenting high fuel load areas while maintaining availability and connectivity of faunal habitat. Applied Mathematical Modelling 54, 298–310.

Rashidi, E., Medal, H., Hoskins, A., 2018a. An attacker-defender model for analyzing the vulnerability of initial attack in wildfire suppression. Naval Research Logistics (NRL) 65, 120–134.

Rashidi, E., Medal, H.R., Hoskins, A., 2018b. Mitigating a pyro-terror attack using fuel treatment. IISE Transactions 50, 499–511.

Salmeron, J., Wood, K., Baldick, R., 2004. Analysis of electric grid security under terrorist threat. IEEE Transactions on power systems 19, 905–912.

Scaparra, M.P., Church, R.L., 2008. A bilevel mixed-integer program for critical infrastructure protection planning. Computers & Operations Research 35, 1905–1923.

Sherali, H.D., Adams, W.P., 2013. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. volume 31. Springer Science & Business Media.

Snyder, L.V., 2006. Facility location under uncertainty: a review. IIE transactions 38, 547–564.

The England & Wales Wildfire Forum, 2019. EWWF 2019 Wildfire Conference. URL: https://www.southwales-fire.gov.uk/app/uploads/2019/11/EWWF-Wildfire-Conference-programe-2019-amended-18-Nov-2019.pdf. accessed February 15, 2021.

The University of Sydney, 2020. More than one billion animals killed in Australian bushfires. URL: https://www.sydney.edu.au/news-opinion/news/2020/01/08/australian-bushfires-more-than-one-billion-animals-impacted.html. accessed February 15, 2021.

Vicente, L., Savard, G., Júdice, J., 1994. Descent approaches for quadratic bilevel programming. Journal of Optimization Theory and Applications 81, 379–399.

Von Stackelberg, H., 2010. Market structure and equilibrium. Springer Science & Business Media.

Wei, Y., Long, Y., 2014. Schedule fuel treatments to fragment high fire hazard fuel patches. Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS) 6, 1–10 (10).

Wei, Y., Rideout, D., Kirsch, A., 2008. An optimization model for locating fuel treatments across a landscape to reduce expected fire losses. Canadian Journal of Forest Research 38, 868–877.

Zhuang, J., Bier, V.M., 2007. Balancing terrorism and natural disasters—defensive strategy with endogenous attacker effort. Operations Research 55, 976–991.