MODELS AND ALGORITHMS FOR WILDFIRE SUPPRESSION PROBLEMS

INSTITUTO DE INFORMÁTICA UFRGS

Gustavo Delazeri

June 2025

- Wildfires are becoming more frequent worldwide.

nature ecology & evolution

Brief Communication

https://doi.org/10.1038/s41559-024-02452-2

Increasing frequency and intensity of the most extreme wildfires on Earth

Received: 22 January 2024
Accepted: 26 May 2024
Published online: 24 June 2024
Check for updates

Calum X. Cunningham, Grant J. Williamson & David M. J. S. Bowman

Climate change is exacerbating wildfire conditions, but evidence is lacking for global trends in extreme fire activity itself. Here we identify energetically extreme wildfire events by calculating daily clusters of summed fire radiative power using 21 years of satellite data, revealing that the frequency of extreme events (≥99.99th percentile) increased by 2.2-fold from 2003 to 2023, with the last 7 years including the 6 most extreme. Although the total area burned on Earth may be declining, our study highlights that fire behaviour is worsening in several regions—particularly the boreal and temperate conifer biomes—with substantial implications for carbon storage and human exposure to wildfire disasters.

- Combating wildfires is expensive.

agênciaBrasil 35 anos

A+ A-

LATEST NEWS | CULTURE ECONOMY EDUCATION GENERAL HEALTH HUMAN RIGHTS INTERNATIONAL JUSTICE POLITICS SPORTS ENVIRONMENT

Politics

Government allocates BRL 514 million to tackle forest fires

The funds come from a special budget authorized by the Supreme Court

WELLTON MÁXIMO
Published on 18/09/2024 - 11:39
Brasília

© MAYANGDI INZAULGARAT/IBAMA

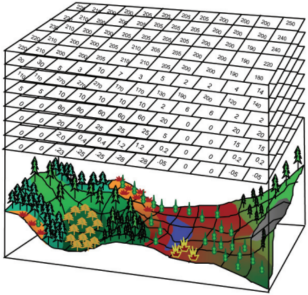
- Decision support tools are needed to help fire managers.
- In this dissertation: how to allocate a set of fire suppression resources to minimize the area burned by a wildfire?

1. A better MIP formulation.
2. An iterated beam search algorithm to solve the problem.
3. A cut-based heuristic to solve the problem.
4. A new instance generator for the problem.
5. A comprehensive computational study comparing the proposed methods with state-of-the-art algorithms.

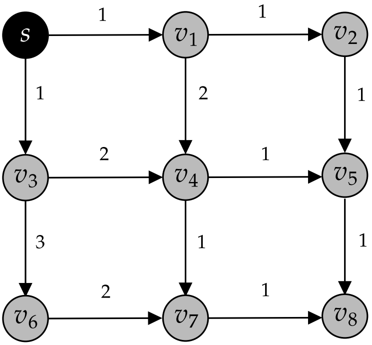
1. Problem definition.
2. Related work.
3. Mixed-integer programming formulation.
4. Iterated Beam Search.
5. Cut-based heuristic.
6. Experimental evaluation 1.
7. New instance generator.
8. Experimental evaluation 2.
9. Conclusion and future work.

Problem Definition

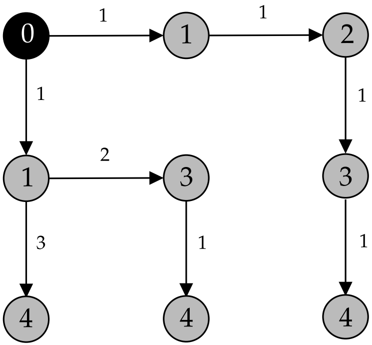
- Landscape
  - Terrain is divided into cells.
  - Each cell has data on vegetation, moisture, slope, wind, etc.
  - Fire spread is modeled based on these data.



- Graph $G = (V, A)$.
- Fire propagation time t_{uv} on each arc $uv \in A$.



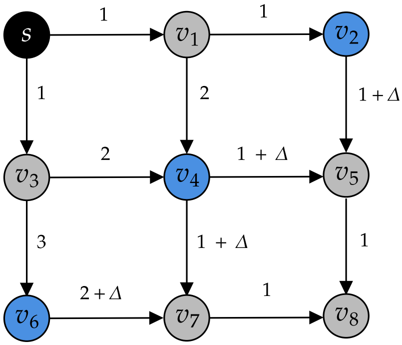
- Ignition vertex $s \in V$.
- Fire propagation is modeled as the shortest path tree from the ignition vertex s.



THE PROBLEM

- Set of fire suppression resources R (e.g., fire crews).
- Release times t_i for each resource $i \in R$.
- Delay Δ caused by resources.

- Resources can be allocated to vertices in the graph.
- Resources add a delay Δ on outgoing arcs.
- Burned vertices cannot receive resources.



- Time horizon H.

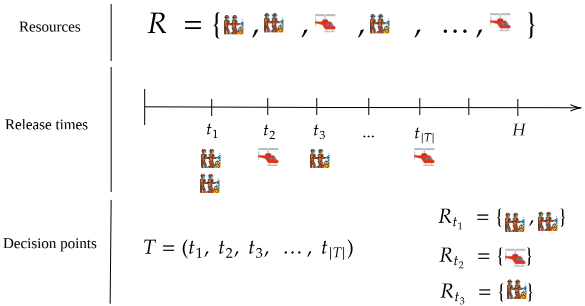
- **Objective**: Find the resource allocation that minimizes the number of vertices that burn before time H.

NOTATION - ALLOCATION OF RESOURCES

- An allocation of resources is a (partial) mapping $\Lambda : R \to V$.
- If $\Lambda_i = v$ for some $i \in R$, then v is *protected* by allocation Λ.
- The empty allocation (no resources allocated) is denoted by Λ_0.
- P^Λ is the set of vertices protected by allocation Λ ($P^{\Lambda_0} = \emptyset$).

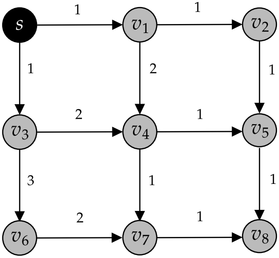
NOTATION - DECISION POINTS

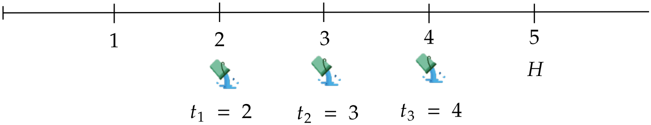
- $T = (t_1, t_2, \ldots, t_{|T|})$: increasing sequence of release times of resources.
- Each $t \in T$ is a **decision point** in the problem.
- $R_t = \{i \in R : t_i = t\}$: set of resources released at time $t \in T$.



EXAMPLE

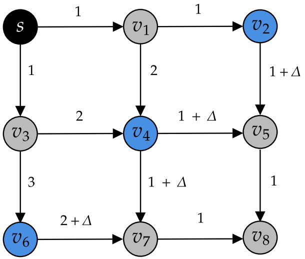
$R = \{1, 2, 3\}$
$\Delta = 2$
$H = 5$





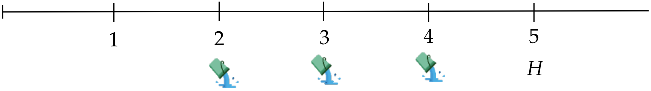
EXAMPLE

$R = \{1, 2, 3\}$
$\Delta = 2$
$H = 5$



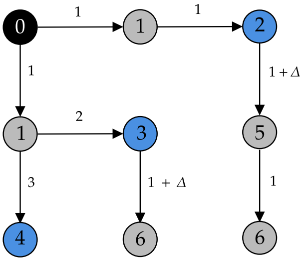
Optimal allocation Λ

$\Lambda_1 = v_2$
$\Lambda_2 = v_4$
$\Lambda_3 = v_6$



$t_1 = 2$ $t_2 = 3$ $t_3 = 4$

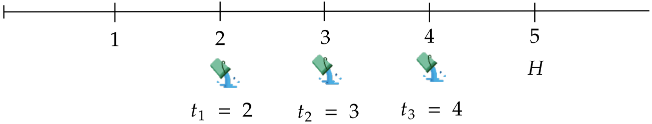
$R = \{1, 2, 3\}$
$\Delta = 2$
$H = 5$



Optimal allocation Λ

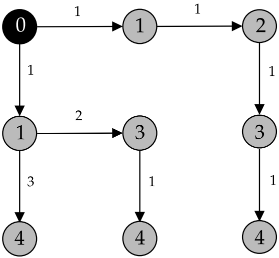
$$\Lambda_1 = v_2$$
$$\Lambda_2 = v_4$$
$$\Lambda_3 = v_6$$

Objective value: 6



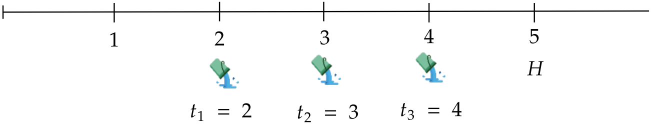
EXAMPLE

$R = \{1, 2, 3\}$
$\Delta = 2$
$H = 5$



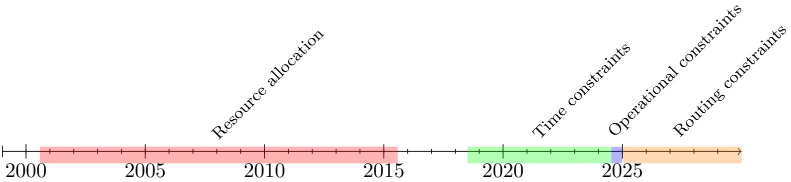
Empty allocation Λ_0

Objective value: 9



Related work

TIMELINE OF CONTRIBUTIONS



- Mendes et al. (2023).
  - MIP formulation.
  - Iterated Local Search.
  - First set of instances.
- Harris et al. (2023).
  - Logic-based Benders Decomposition.
  - Harder instances.

And so our journey begins...

Mixed-Integer Programming Formulation

MODEL STRUCTURE

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

$$\boxed{\text{Fire propagation}}$$

$$\boxed{\text{Resource allocation}}$$

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0,1\}, \qquad v \in V,$$

$$r_{tv} \in \{0,1\}, \qquad t \in T, v \in V.$$

MODEL VARIABLES

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

Fire arrival time at vertex v

$$y_v \in \{0, 1\}, \qquad v \in V,$$

1 if vertex v burned before time H

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

1 if vertex v received a resource at time t

MODEL CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$ Minimize the number of burned vertices

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$ if $a_v < H$ then $y_v = 1$

$$\boxed{\text{Fire propagation}}$$

$$\boxed{\text{Resource allocation}}$$

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0,1\}, \qquad v \in V,$$

$$r_{tv} \in \{0,1\}, \qquad t \in T, v \in V.$$

RESOURCE CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

Resource allocation

$$\sum_{v \in V} r_{tv} \leq |R_t|, \qquad t \in T,$$

$$\sum_{t \in T} r_{tv} \leq 1, \qquad v \in V,$$

$$a_v \geq r_{tv} \cdot t, \qquad v \in V,\ t \in T.$$

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0,1\}, \qquad v \in V,$$

$$r_{tv} \in \{0,1\}, \qquad t \in T, v \in V.$$

FIRE PROPAGATION CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

$$-\sum_{v \in N_s^+} x_{sv} = -|V| + 1,$$

$$-\sum_{w \in N_v^+} x_{vw} + \sum_{u \in N_v^-} x_{uv} = 1, \qquad v \in V \setminus \{s\},$$

$$x_{uv} \leq (|V| - 1) q_{uv}, \qquad uv \in A,$$

$$s_{uv} \leq M(1 - q_{uv}), \qquad uv \in A,$$

$$a_s = 0$$

$$a_v - a_u + s_{uv} = t_{uv} + \Delta \sum_{t \in T} r_{tu}, \qquad uv \in A,$$

$$s_{uv} \geq 0, \qquad uv \in A,$$

$$x_{uv} \geq 0, \qquad uv \in A,$$

$$q_{uv} \in \{0, 1\}, \qquad uv \in A,$$

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0, 1\}, \qquad v \in V,$$

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

FIRE PROPAGATION CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

$$-\sum_{v \in N_s^+} x_{sv} = -|V| + 1,$$

$$-\sum_{w \in N_v^+} x_{vw} + \sum_{u \in N_v^-} x_{uv} = 1, \qquad v \in V \setminus \{s\},$$

$$x_{uv} \leq (|V| - 1) q_{uv}, \qquad uv \in A,$$

$$s_{uv} \leq M(1 - q_{uv}), \qquad uv \in A,$$

$$a_s = 0$$

$$a_v - a_u + s_{uv} = t_{uv} + \Delta \sum_{t \in T} r_{tu}, \qquad uv \in A,$$

$\boxed{s_{uv} \geq 0,}$ Slack of each arc. $uv \in A,$

$\boxed{x_{uv} \geq 0,}$ Flow pushed through each arc. $uv \in A,$

$\boxed{q_{uv} \in \{0, 1\},}$ 1 if arc uv is used. $uv \in A,$

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0, 1\}, \qquad v \in V,$$

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

FIRE PROPAGATION CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0,1\}, \qquad v \in V,$$

$$r_{tv} \in \{0,1\}, \qquad t \in T, v \in V.$$

Flow constraints

$$-\sum_{v \in N_s^+} x_{sv} = -|V| + 1,$$

$$-\sum_{w \in N_v^+} x_{vw} + \sum_{u \in N_v^-} x_{uv} = 1, \qquad v \in V \setminus \{s\},$$

$$x_{uv} \leq (|V| - 1) q_{uv}, \qquad uv \in A,$$

$$s_{uv} \leq M(1 - q_{uv}), \qquad uv \in A,$$

$$a_s = 0$$

$$a_v - a_u + s_{uv} = t_{uv} + \Delta \sum_{t \in T} r_{tu}, \qquad uv \in A,$$

$$s_{uv} \geq 0, \qquad uv \in A,$$

$$x_{uv} \geq 0, \qquad uv \in A,$$

$$q_{uv} \in \{0,1\}, \qquad uv \in A,$$

FIRE PROPAGATION CONSTRAINTS

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v / H, \qquad v \in V,$$

Fire propagation

$$-\sum_{v \in N_s^+} x_{sv} = -|V| + 1,$$

$$-\sum_{w \in N_v^+} x_{vw} + \sum_{u \in N_v^-} x_{uv} = 1, \qquad v \in V \setminus \{s\},$$

Linking constraints

$$x_{uv} \leq (|V| - 1) q_{uv}, \qquad uv \in A,$$

$$s_{uv} \leq M(1 - q_{uv}), \qquad uv \in A,$$

$$a_s = 0$$

$$a_v - a_u + s_{uv} = t_{uv} + \Delta \sum_{t \in T} r_{tu}, \qquad uv \in A,$$

$$s_{uv} \geq 0, \qquad uv \in A,$$

$$x_{uv} \geq 0, \qquad uv \in A,$$

$$q_{uv} \in \{0, 1\}, \qquad uv \in A,$$

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0, 1\}, \qquad v \in V,$$

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

$$\min. \sum_{v\in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0,1\}, \qquad v \in V,$$

$$r_{tv} \in \{0,1\}, \qquad t \in T, v \in V.$$

$$-\sum_{v\in N_s^+} x_{sv} = -|V| + 1,$$

$$-\sum_{w\in N_v^+} x_{vw} + \sum_{u\in N_v^-} x_{uv} = 1, \qquad v \in V \setminus \{s\},$$

$$x_{uv} \leq (|V|-1)q_{uv}, \qquad uv \in A,$$

$$s_{uv} \leq M(1-q_{uv}), \qquad uv \in A,$$

Fire arrival time constraints

$$a_s = 0$$

$$a_v - a_u + s_{uv} = t_{uv} + \Delta \sum_{t\in T} r_{tu}, \qquad uv \in A,$$

$$s_{uv} \geq 0, \qquad uv \in A,$$

$$x_{uv} \geq 0, \qquad uv \in A,$$

$$q_{uv} \in \{0,1\}, \qquad uv \in A,$$

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v/H, \qquad v \in V,$$

Fire propagation

Resource allocation

$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0, 1\}, \qquad v \in V,$$

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

The objective function pushes a_v to be greater than H.

We only have to impose upper bounds on the fire arrival time.

PROPOSED REFORMULATION

$$\min. \sum_{v \in V} y_v$$

s.t.

$$y_v \geq 1 - a_v / H, \qquad v \in V,$$

Fire propagation 2.0

$$a_s = 0$$

$$a_v \leq a_u + t_{uv} + \Delta \sum_{t \in T} r_{tv} \qquad uv \in A,$$

Resource allocation

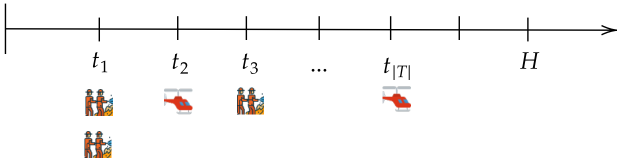
$$a_v \geq 0, \qquad v \in V,$$

$$y_v \in \{0, 1\}, \qquad v \in V,$$

$$r_{tv} \in \{0, 1\}, \qquad t \in T, v \in V.$$

Iterated Beam Search

- At each decision point $t \in T$, we have to decide which vertices will receive the released resources.

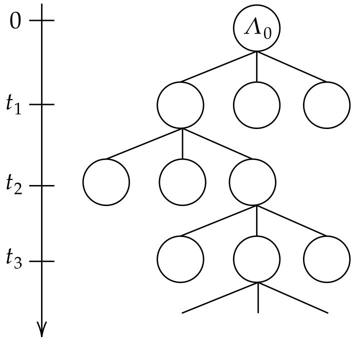


- The problem structure suggests the organization of the search space as a tree.
- Each node in the tree is a partial allocation of resources.

MOTIVATION

- We explore the search space using a beam search strategy.
- Beam search is an incomplete tree search algorithm.

- Start with the empty allocation $\mathcal{A} = \{\Lambda_0\}$.
- For each decision point $t \in T$:
  - Expand each allocation in $\mathcal{A}$ η times using R_t.
  - Select the best β allocations according to h.



- Components of a beam search algorithm:
  - Heuristic function h to prune allocations.
  - Expansion of allocations at each decision point.

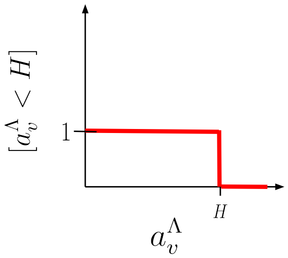
NOTATION - FIRE ARRIVAL TIMES

- An allocation of resources Λ changes fire arrival times.
- We denote by a_v^{Λ} the fire arrival time at vertex $v \in V$ under allocation Λ.

How to evaluate a partial allocation of resources Λ?

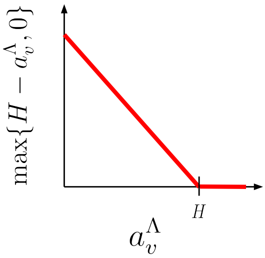
Number of burned vertices

$$h_1(\Lambda) = \sum_{v \in V} [a_v^\Lambda < H]$$



Time to survival

$$h_2(\Lambda) = \sum_{v \in V} \max\{H - a_v^\Lambda, 0\}$$

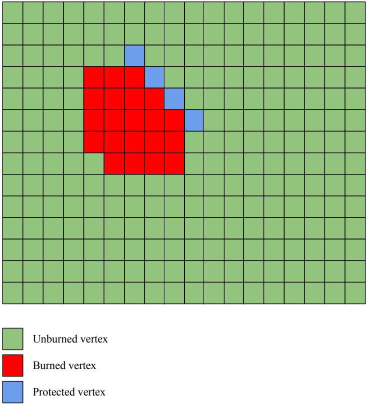


- Heuristic h_1 is misleading in the early stages of the search.
- We start pruning with h_2 and, at some point, switch to h_1.

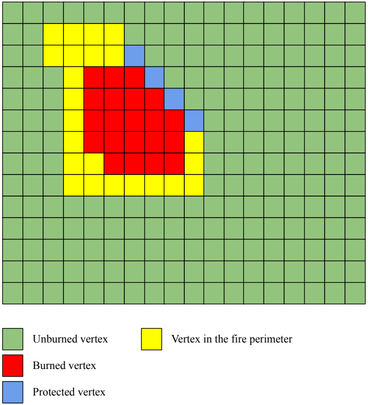
EXPANDING AN ALLOCATION OF RESOURCES

- Given a partial allocation Λ and a set of resources R_t released at time $t \in T$.
- We want to expand Λ by allocating resources in R_t.
- Search space: all combinations of $|R_t|$ unburned vertices. Too large to explore exhaustively.

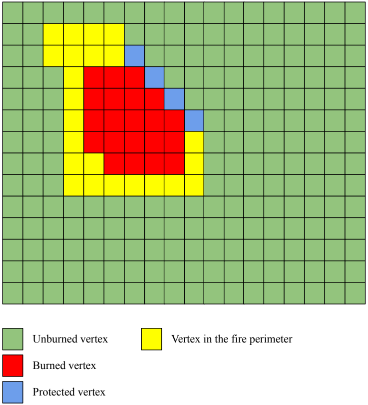
Partial solution Λ:



- Heuristic reduction rule:
  only consider vertices that are in the **fire perimeter**.
- Fire perimeter:
  set of vertices that will burn soon.
- How soon? Read the dissertation.

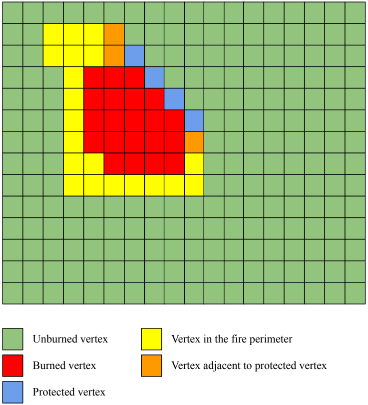


- Let F be the set of vertices in the fire perimeter.
- Assign a probability p_v to each vertex $v \in F$.
- To expand Λ, we sample $|R_t|$ vertices from F according to the probabilities p_v.



EXPANDING AN ALLOCATION OF RESOURCES

- Good solutions usually form fire breaks.
- We want to bias the sampling towards vertices that are close to protected vertices.
- Let $N \subseteq F$ be the set of vertices that are adjacent to protected vertices.



- Let $v \in F$ be any vertex in the fire perimeter

$$p_v \propto \begin{cases} 1 + p^{\frac{|F|-|N|}{|N|}}, & \text{if } v \in N, \\ 1 - p, & \text{if } v \in F \setminus N. \end{cases}$$

- We use N to shift the probability mass towards vertices that are close to protected vertices.
- Parameter $p \in [0, 1]$ controls the bias.

EXPANDING AN ALLOCATION OF RESOURCES

$p = 0$

$$p_v = \frac{1}{|F|}$$

$0 < p < 1$

$$p_v = \begin{cases} \frac{1}{|F|} + p\left(\frac{1}{|N|} - \frac{1}{|F|}\right), & v \in N, \\ \frac{1}{|F|} - p\frac{1}{|F|}, & v \in F \setminus N. \end{cases}$$

$p = 1$

$$p_v = \begin{cases} \frac{1}{|N|}, & v \in N, \\ 0, & \text{otherwise.} \end{cases}$$

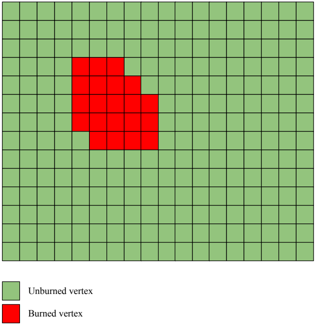
Cut-based Heuristic

MOTIVATION

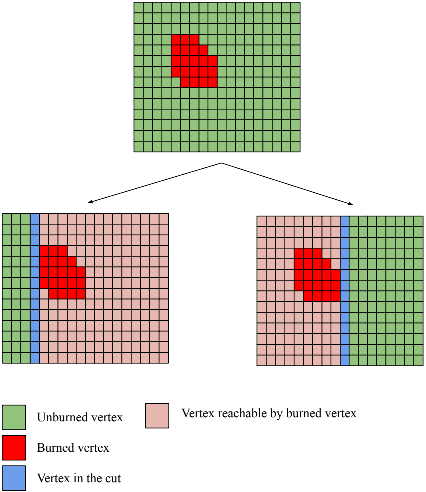
- Building firebreaks is a common strategy to suppress fire.
- Beam search tries to build firebreaks indirectly by sampling vertices in the fire perimeter.
- We can build firebreaks directly by solving a vertex cut problem.

SETUP

- Assume that all resources are released at time t.
- Let B be the set of vertices that burn before time t.



- A fire break is a vertex cut $P \subseteq V$ that separates the vertices in B from the rest of the graph.
- Find the vertex cut that minimizes the vertices reachable by burned vertices.



- Assume that all resources are released at time t and let B be the set of vertices that burn before time t.
- Solve M_{cut} to find a vertex cut $P \subseteq V$ that separates vertices in B from the rest of the graph.

$$
\begin{aligned}
(M_{cut}) \quad & \min. \quad \sum_{v \in V} y_v && && \text{(C.1)} \\
& \text{s.t.} \quad y_v = 1 && v \in B && \text{(C.2)} \\
& \quad x_v = 0 && v \in B && \text{(C.3)} \\
& \quad y_u \leq y_v + x_u && uv \in A && \text{(C.4)} \\
& \quad \sum_{v \in V} x_v \leq |R| && && \text{(C.5)} \\
& \quad x_v, y_v \in \{0, 1\} && v \in V && \text{(C.6)}
\end{aligned}
$$

Given a set of vertices $P \subseteq V$ that form a vertex cut, how to build an allocation of resources?

- Greedy rule: assign the available resource with the earliest release time to the vertex in P with the earliest fire arrival time.
- Proposition: there exists an allocation Λ such that $P^{\Lambda} = P$ if, and only if, the greedy rule can be applied to all vertices in P.

- How to choose t?
- Smaller t means the burned area is smaller, resulting in better cuts. However, most cuts are not feasible allocations.
- Let t_1 and $t_{|T|}$ be the first and last time instants when resources are released, respectively.
- Idea:
  - Binary search for the smallest $t \in [t_1, t_{|T|}]$ such that the vertex cut P can be transformed into a feasible allocation.
  - Return the best feasible allocation found during the search.

Experiments

- How do the proposed algorithms compare with the approaches available in the literature
  - in terms of solution quality?
  - in terms of finding optimal solutions?

- Instances:
  - 16 instances from the literature.
- Algorithms:
  - MIP formulation 1 (this work).
  - Iterated Beam Search (this work).
  - Cut-based Heuristic (this work).
  - MIP formulation 2 (Mendes et al. (2023)).
  - Iterated Local Search (Mendes et al. (2023)).
  - Logic-based Benders Decomposition (Harris et al. (2023)).
  - Random Search.
- Protocol:
  - 10 replications.
  - Time limit of 600 seconds.
  - All algorithms executed in the same environment.

INSTANCES

- 20x20 grids.
- Optimal solutions are known.

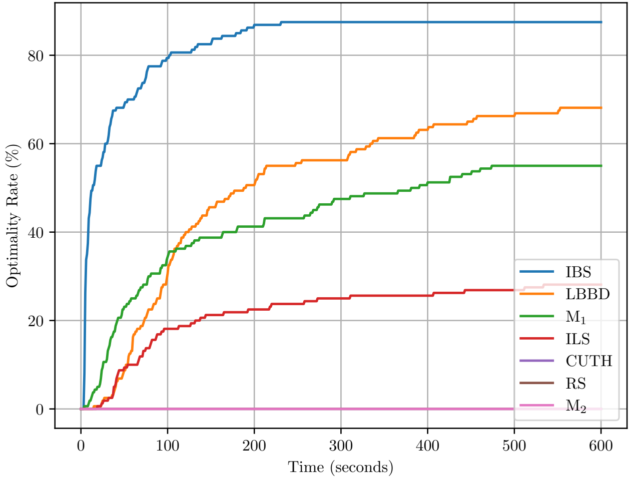
- Code:
  - IBS: Iterated Beam Search.
  - CUTH: Cut-based Heuristic.
  - LBBD: Logic-based Benders Decomposition.
  - ILS: Iterated Local Search.
  - RS: Random Search.
  - M_1: Proposed MIP formulation.
  - M_2: MIP formulation from the literature.

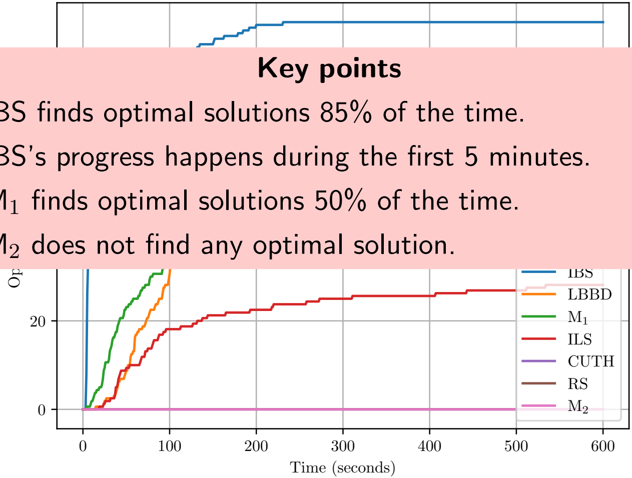
Avg. absolute deviation to optimal objective						
IBS	CUTH	M_1	M_2	ILS	LBBD	RS
0.1	49.2	1.8	59.7	6.2	2.0	61.5

Key points

- IBS finds the optimal solution of 14 out of the 16 instances (in all 10 replications).
- IBS has the smallest average absolute mean deviation.
- M_1 is far superior to M_2.
- M_1 obtains better solutions than LBBD.
- CUTH is not competitive with the other algorithms (more on that later).

Avg. absolute deviation to optimal objective						
IBS	CUTH	M_1	M_2	ILS	LBBD	RS
0.1	49.2	1.8	59.7	6.2	2.0	61.5


Optimality Rate (%)
0
20
40
60
80
0
100
200
300
400
500
600
Time (seconds)
IBS
LBBD
M_1
ILS
CUTH
RS
M_2



Key points

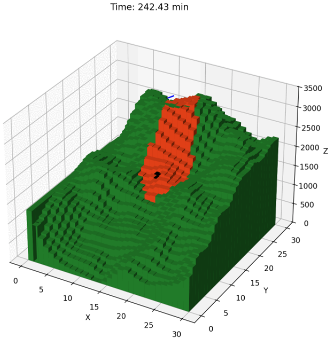
- IBS finds optimal solutions 85% of the time.
- IBS's progress happens during the first 5 minutes.
- M_1 finds optimal solutions 50% of the time.
- M_2 does not find any optimal solution.

Instance Generator

- Problems with literature benchmarks
  1. Small number of instances.
  2. Lack of physical realism.
  3. No systematic process to generate instances.
  4. No control over instance characteristics.

- Fire propagation using Rothermel's model.
- Natural-looking landscapes using Perlin noise.
- 8 experimental factors:
  - Grid size.
  - Decision points.
  - Delay.
  - Number of resources.
  - Slope.
  - Wind.
  - First release time.
  - Last release time.

$$R = \frac{I_R \xi}{\rho_b \epsilon Q_{ig}}(1 + \Phi_w + \Phi_s)$$



Experiments

- In which scenario each algorithm performs best (or worse)?

- Instances:
  - Pairwise factor variation (next slide).
- Algorithms:
  - Same as before, but now we drop the old MIP formulation.
- Protocol:
  - Same as before, but now the time limit varies with the grid size.

Group	Parameter	Default Value
Instance Size	Grid size	30×30
	Decision points	Medium
Suppression Capacity	Delay	High
	# of resources	Moderate
Environment	Slope	Moderate
	Wind	Light
Release Window	First release time	Early
	Last release time	Very late

- Environmental factors and release window have a negligible impact on the results.
- Instance size and suppression capacity affect algorithm performance considerably. We focus on these two groups.

- Experiment:
  - Vary grid size from 20×20 to 80×80.
  - Vary decision points from 5 to 20.
  - Keep other factors constant.
- Performance metrics:
  - Relative deviation to BKV.
  - Percentage of vertices saved by the best known solution (BKV).

		Average Relative Gap to BKV (%)						Saved Vertices (%)
$n \times n$	$\lvert T \rvert$	CUTH	IBS	ILS	LBBD	M_1	RS	
20×20	5	30.9	**3.1**	24.9	15.9	10.3	33.3	29.5
	10	32.6	**0.0**	18.7	30.1	19.3	38.1	31.8
	20	34.1	**0.0**	29.1	36.1	22.2	40.4	32.5
30×30	5	23.4	17.3	29.2	26.3	**14.4**	29.6	25.8
	10	33.7	**0.6**	44.0	38.7	26.7	41.0	31.7
	20	44.6	**0.0**	58.5	58.7	37.8	54.1	37.0
40×40	5	17.1	26.8	30.4	29.9	**10.7**	27.1	23.4
	10	29.7	**1.2**	44.4	44.2	30.7	41.2	30.8
	20	31.0	**0.0**	46.0	46.1	32.9	42.5	31.6
80×80	5	**2.6**	8.1	8.9	8.9	3.3	8.5	8.2
	10	**0.0**	5.7	7.4	7.3	3.2	7.0	6.9
	20	30.8	**12.2**	40.7	40.8	36.8	40.4	29.0

		Average Relative Gap to BKV (%)						Saved
	5	**2.6**	8.1	8.9	8.9	3.3	8.5	8.2
80×80	10	**0.0**	5.7	7.4	7.3	3.2	7.0	6.9
	20	30.8	**12.2**	40.7	40.8	36.8	40.4	29.0

Key points

- IBS is the best algorithm in most instances.
- IBS performance degrades in instances with few decision points.
- LBBD does not scale well with instance size.
- CUTH obtains a similar performance to M_1 as instance size increases.
- The best solutions save $\approx 30\%$ of the vertices in the grid.

- Experiment:
  - Vary delay Δ from low to high
  - Vary $\#$ of resources from few to many
  - Keep other factors constant.
- Performance metrics:
  - Same as before.

		Average Relative Gap to BKV (%)						Saved Vertices (%)
# of resources	Delay	CUTH	IBS	ILS	LBBD	M_1	RS	
Few	Low	**0.1**	5.4	4.0	3.0	0.5	4.2	5.8
	Medium	**0.5**	5.8	2.9	4.6	2.0	4.6	6.1
	High	**4.3**	9.8	6.6	9.3	5.5	8.6	9.6
Moderate	Low	5.8	5.5	12.4	12.1	**2.4**	9.2	11.7
	Medium	10.9	**0.6**	19.9	13.3	8.3	17.0	17.7
	High	33.7	**0.6**	44.0	38.1	26.7	41.0	31.7
Many	Low	66.7	10.3	75.3	77.4	**6.4**	69.4	44.0
	Medium	91.2	**21.6**	119.6	120.8	22.3	114.1	55.8
	High	125.1	**7.8**	157.7	129.5	17.0	152.1	62.4

		Average Relative Gap to BKV (%)						Saved Vertices (%)
# of resources	Delay	CUTH	IBS	ILS	LBBD	M_1	RS	
	Low	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Many	Medium	91.2	**21.6**	119.6	120.8	22.3	114.1	55.8
	High	125.1	**7.8**	157.7	129.5	17.0	152.1	62.4

Key points

- IBS performance degrades in instances with low delay or few resources.
- CUTH obtains the best results when the suppression capacity is very low.

Conclusion

CONTRIBUTIONS

1. An improved MIP formulation for the Fire Suppression Problem.
2. A Beam Search algorithm that outperforms the state-of-the-art.
3. A Cut-based Heuristic that is competitive in large instances.
4. A new instance generator that allows for systematic generation of instances with different characteristics.
5. An extended model that incorporates operational constraints and characteristics of fire suppression resources (not covered here).

EvoApps 2024

Delazeri, G. & Ritt, M. (2024).
Iterated Beam Search for Wildland Fire Suppression.
In *Applications of Evolutionary Computation*
(pp. 273-286).

SBPO 2024

Delazeri, G. & Ritt, M. (2024).
A General Model for Wildfire Suppression Problems.
In *Anais do Simpósio Brasileiro de Pesquisa Operacional*
(SBPO 2024, Vol. 56).

- Improving beam search:
  - IBS has its weaknesses. Can we fix them?
  - IBS iterations are (almost) independent. Can we learn from previous iterations?
- Better exact algorithms.


"That's all Folks!"