Data Instance generator and optimization models for evacuation planning in the event of wildfire

Christian Artigues[1] Emmanuel Hébrard[1] Yannick Pencolé[1]
Andreas Schutt[2] Peter J. Stuckey[3]

[1]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[2]Decision Sciences, Data61, CSIRO, Australia
[3]Department of Computing and Information Systems,
The University of Melbourne, Victoria 3010, Australia

Abstract

One critical part of decision support during the response phase to a wildfire is the ability to perform large-scale evacuation planning. While in practice most evacuation planning is principally designed by experts using simple heuristic approaches or scenario simulations, more recently optimization approaches to evacuation planning have been carried out, notably in the context of floodings. Evacuation planning in case of wildfires is much harder as wildfire propagations are inherently less predictable than floods. This paper present a new optimization model for evacuation planning in the event of wildfire aiming at maximizing the temporal safety margin between the evacuees and the actual or potential wildfire front. As a first contribution, an open-source data instance generator based on road network generation via quadtrees and a basic fire propagation model is proposed to the community. As a second contribution we propose 0–1 integer programming and constraint programming formulations enhanced with a simple compression heuristic that are compared on 240 problem instances build by the generator. The results show that the generated instances are computationally challenging and that the contraint programming framework obtains the best performance.

In: G. Di Stafano, A. Navarra Editors: Proceedings of the RSFF'18 Workshop, L'Aquila, Italy, 19-20-July-2018, published at http://ceur-ws.org

1 Introduction

The overall objective of the GEO-SAFE project [8] is to develop methods and tools enabling to set up an integrated decision support system to assist authorities in optimizing the resources during the response phase to a wildfire (fire suppression, life and goods protection). One critical and crucial part of this integrated decision support is the ability to perform large-scale evacuation planning [15]. While in practice most evacuation planning is principally designed by experts using simple heuristic approaches or scenario simulations [17], more recently optimization approaches to evacuation planning have been addressed, using a variety of optimization technology, as surveyed recenly in [2]. This paper presents a challenging variant of the evacuation planning problem in case of wild fire issued from exchanges with practitioners in the context of the GEO-SAFE project and from a specific literature review. A large amount of work has been carried out, notably at NICTA[1],[11, 4, 12, 10, 5, 6, 9, 13] mainly in the context of floodings, which can be transposed under some adaptations to evacuation in case of fires. Evacuation planning in case of wildfires is indeed much harder. Wildfire propagations are inherently less predictable than floods. While flood levels mostly rely on the fixed topology of the area and rainfalls, wildfire mainly depends on the wildland fuels [14, 1], on the slope of the burning ground and more importantly on the speed and direction of the wind that can suddenly change at any time [18, 16]. Therefore, evacuation planning dedicated to wildfires must be much more robust to difference future scenarios. A good evacuation plan in case of wildfire must not only minimize the evacuation time of the population but also maximize the spatial and temporal safety margin between the evacuees and the actual or potential wildfire front.

This paper present a new optimization model for evacuation planning in the event of wildfire as well as a problem instance generator. On these instances, basic 0–1 integer programming and constraint programming formulations enhanced with a simple compression heuristic are compared. In Section 2, we provide a literature review and we define the considered problem. Section 3 presents the instance generator. The basic 0–1 integer programming, constraint programming formulations and the heuristic are proposed in Section 4. Computational experiments are given in Section 5.

2 State-of-the-art review and problem definition

2.1 Basic evacuation data

We adopt the notation an terminology given in [6]. There is a directed graph $\mathcal{G}(\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ representing e.g. a road network in a region that must be evacuated. The graph is made of:

- the set of *evacuation* nodes $\mathcal{E}$. An evacuation node represents a zone where people to be evacuated are regrouped),
- the set of *safe* nodes $\mathcal{S}$. A safe node represent a safe geographical zone that people located in the evacuation nodes must reach during the planning horizon
- the set of *transit* nodes $\mathcal{T}$. A transit node represent an intersection in the road network that can be traversed by the vehicles carrying evacuated people from the evacuation zones to the safe zones.

[1]Former Australia's Information and Communications Technology Research Centre

Suppose the time is discretized in minutes. Each evacuation node $k \in \mathcal{E}$ is associated with a number of evacuees d_k. Each arc $q \in \mathcal{A}$ has a capacity u_e in evacuees/minute, a travel time t_e giving the number of minutes that a vehicle takes to traverse the arc and a end time b_e, which gives the time at which the arc becomes unavailable due to the fire propagation.

Other characteristics may appear in the variants considered in the cited papers.

2.2 Previously considered evacuation problems

We limit our review to the the NICTA papers as they explain the genesis of the final model w.r.t. practical considerations. The first considered problem and solution methods were presented by Pillac *et al.* in 2013 in a research report that was later published in [11]. One of the practical motivation of the work was that in an urgency situation, the possibility of choices in a road network, such as a fork, generate stress among the population. This is why it is preferable to define a single evacuation path for each evacuation zone. Hence the problem considered in [11] can be described as a two-level optimization problem, which can be solved in an integrated way or by a two-phase approach. At the first level, exactly one evacuation path is determined from each evacuation node to a single safe node. At the second level the flow of evacuees is scheduled trough the path. A MILP (called the restricted flow model RF) is proposed to solve the problem in an integrated way. It is based on time-expanded graph, in which each node is duplicated for each time period. Arc capacities u_e in the time expanded network ensure that the traffic does not exceed the road capacity, especially when several paths use the same arc. The model includes continuous flow variable per evacuation node per arc and in the expanded network and also a binary variable per edge in the (non-expanded) graph $\mathcal{G}$ to ensure the uniqueness of the path for each evacuation node. As the MILP is intractable on a set of instances (HN) derived from a real case scenario involving 70000 evacuees in the Hawkesbury-Nepean floodplain, located North-West of Sydney, an iterative column-generation-like two-phase heuristic is proposed. Given a set of potential paths, a master problem solves the path selection and flow scheduling problem with another MILP (called the conflict-based heuristic path generation master problem, CPG-MP) involving a reduced number of binary constraints (one per path). A sub-problem finds new paths based on a subset of critical nodes by solving a multiple-origin, multiple-destination shortest path problem. A lexicographic objective is considered. The main objective is to maximize the number of evacuated people during the time horizon, while a secondary objective aims at maximizing the evacuation start time, based on the practical relevance. Note that the second objective is only indirectly tackled by weighting each arc in the time-expanded graph by a cost inversely proportional to the arc time, hence solving a min-cost flow problem.

Subsequent papers describe variants of the models and new methods. A more sophisticated variant of the heuristic was proposed in Pillac *et al.* (2014) [12] for the same problem with, however, a different secondary objective for the min-cost flow, aiming at minimizing the evacuation time (called the clearance time), which is indirectly obtained by weighting the arc in the time-expanded graph by a cost proportional to the arc time.

In Even *et al.* (2014) [4], an additional degree of freedom is introduced, giving the possibility of selecting contraflow roads, which consists in reversing the direction of some major roads. In practical cases, this possibility can highly increase the network capacity for evacuation. This is done by introducing a subset $\mathcal{A}_c$ of arcs that can be used in contraflows and by modifying the CPG-MP, interestingly without the need of introducing additional binary variables. A software called the NICTA Evacuation Planner is also presented, with new instances with up to 1 000 000 evacuees, which are solved requiring up to 30 min. of CPU time.

In Even *et al.* (2015)[5], the case where the selected paths must form a convergent subgraph, i.e. for which each node has at most one outgoing arc in the set of paths, is considered. This is motivated by the fact that convergent evacuation paths can be better controlled. Indeed, even if in the previous model, a single path was assigned to each evacuation node, in the case where two paths are merged and then are separated again, a driver can be confronted to a choice and take a decision that would not correspond to what was planed. Hence the set of paths now form a tree rooted at the safe node. The solution method is still a two-phase method but does not follow anymore the column generation principle. A tree is first built by a MIP working on an aggregated graph (without time discretization). The second phase is the flow scheduling problem given the computed tree, which is a maximum flow problem. The first phase is now able to produce good upper bounds on the maximum number of evacuees and on the minimum clearance time. A simulation study shows that the convergent model outperforms the general model when the presence of a fork in the network generated an hesitation for the driver that is captured by a 0.75 second delay.

This model is further developed by Kumar *et al.* (2016) [9] to incorporate network design aspects in the problem to model possible infrastructure enhancement decisions, as for the west Sydney case concerns about the capacity of the road network were expressed. Two additional decisions are introduced : adding lanes to a road (arc) and elevating a road (arc). The first decision results in an increase of the capacity of the arc while the second decision postpones the flooding time, both coming at a cost. These decisions can be incorporated in the tree design problem, which is also the master problem of a Benders decomposition method. The objective is still to maximize the number of evacuated people under a budget constraint that limits the infrastructure upgrades. The maximum flow subproblem is used to obtain a feasible flow schedule and also to generate optimality cuts that are reinjected in the master problem. To increase the performance of the Benders method, pareto-optimal cuts are generated.

All previous approaches assumed that each individual vehicle of an evacuation (flow unit) moves independently from the other vehicles of the same zone, and exactly as prescribed by the maximum flow model. Pillac *et al.* [10] propose to use the concept of response curves to incorporate behavioral models in the problem. In practice, instead of assigning a start time to each evacuee, the authorities can influence the evacuation start time of a zone and the level of resources mobilized to increase the evacuation rate (e.g. number of agents knocking on people's door), to which people answer according to a behavioral model abstracted by a response curve. It follows that to each evacuation zone $k \in \mathcal{E}$, a set of response curves $\mathcal{F}_k$ is given. For each evacuated zone, a start time δ_k and a response curve $f_f \in \mathcal{F}_k$ has to be assigned such that the flow of evacuees leaving zone k at time t is given by

$$\phi_k^t = \begin{cases} 0 & \text{if } t < \delta \\ f(t-\delta) & \text{if } t \geq \delta. \end{cases}$$

A column generation approach is proposed where the set of all evacuation plans of a zone is considered, where a evacuation plan is a path from the evacuated zone to a safe node, the start time and the response curve. The master problem selects a plan for each evacuation zone in such a way that the network capacity is satisfied and the total cost of the plans is minimized (without ensuring that a convergent is obtained). The plan generation subproblem is solved either via a MIP or via a shortest path approach. The methods experiences difficulties is solving realistic instances due to a long-tail effect.

Another drawback of the flow model is that it generated preemptive evacuation plans. As flow units are routed independently there are time periods in which the evacuation of a zone may be stopped and reinitiated later. Even *et al.* (2015) [6] report that this creates serious implementation issues for the evacuation plans. They propose another model in which the evacuation rate is a decision variable that remains fixed as soon as the evacuation starts and as long as the zone is not fully evacuated or the time horizon is reached. They assume that the evacuation path of each evacuation zone is already determined and concentrate only on the scheduling problem. We present this model in the case that all people must be evacuated and the objective is to minimize the clearance time. Let δ_k denote the start time of evacuation of zone k and let λ_k denote the evacuation rate of zone k. Let $\mathcal{A}_k = (e_k^1, \ldots, e_k^{|\mathcal{A}_k|})$ the evacuation path of zone k given as its list of edges. An evacuation plan defines a task J_k^i for each edge e_k^i with a start time $S_k^i = \delta_k + \sum_{q=1}^{i-1} t_k^q$ where t_k^i is the travel time of edge e_k^i. The evacuation has a variable duration p_k and the total number of evacuees is $p_k\lambda_k$ with $p_k\lambda_k = d_k$ since all people must be evacuated. Each edge task $e_k^i \in \mathcal{A}_k$ has duration p_k. Let $\mathcal{J}_e$ the set of tasks that use edge $e \in \mathcal{A}$. We also denote as u_k the maximum evacuation rate of an evacuation node k, which can be seen as a node capacity. The model can be written as follows :

$$\min C_{\max} \tag{1}$$

$$\text{s.t.}$$

$$C_{\max} \geq S_k^{|\mathcal{A}_k|} + p_k \qquad \forall k \in \mathcal{E} \tag{2}$$

$$S_k^i + p_k \leq b_{e_k^i} \qquad \forall k \in \mathcal{E}, \forall i = 1, \ldots, |\mathcal{A}_k| \tag{3}$$

$$p_k\lambda_k = d_k \qquad \forall k \in \mathcal{E} \tag{4}$$

$$S_k^i = \delta_k + \sum_{q=1}^{i-1} t_k^q \qquad \forall k \in \mathcal{E}; \forall i = 1, \ldots, |\mathcal{A}_k| \tag{5}$$

$$\sum_{J_k^i \in \mathcal{J}_e, S_k^i \leq t < S_k^i + p_k} \lambda_k \leq u_e \qquad \forall e \in \mathcal{A}, \forall t \geq 0 \tag{6}$$

$$\delta_k \geq 0 \qquad \forall k \in \mathcal{E} \tag{7}$$

$$u_k \geq \lambda_k \geq 0 \qquad \forall k \in \mathcal{E} \tag{8}$$

$$p_k \geq 0 \qquad \forall k \in \mathcal{E} \tag{9}$$

This is a no-wait total work- and resource-constrained project scheduling problem where (4) are the total work constraints, (5) are the no-wait constraints, as the start time of the evacuation task on a arc of the evacuation path starts exactly at the decided evacuation time δ_k plus the total travel time along the path toward edge e. Constraints (6) are the capacity constraints on each edge e. The problem was efficiently solved in [6] via constraint programming for both the clearance time minimization version and the maximization of the number of evacuated people variant.

2.3 A new evacuation planning problem in case of wild fires

In the work of Even *et al.* (2015) [6] and previous studies, the time b_e at which arc e becomes unavailable comes from a flood propagation model, which is pretty accurate. In the case of fire, even if precise propagation models can be obtained, they depend on multiple parameters. Among them, the wind has a great variability. The subject of promising further studies would be to consider explicitly uncertainty via robust of stochastic approaches, but this would require the definition of various scenarios possibly associated with probability distributions. An alternative to modeling uncertainty of the unavailability dates would be to consider an objective function that seeks to maximize the length of the time interval $[C_e, b_e]$, where C_e is the completion time of the last task using edge e. It follows that in this paper we consider the following optimization problem, with objective function (10) giving the maximum (possibly negative) slack on each evacuation arc, weighted by the population.

$$\max \min_{k \in \mathcal{E}, \forall i=1,\dots,|\mathcal{A}_k|} d_k(b_{e_k^i} - S_k^i + p_k) \tag{10}$$

$$\text{s.t.}(4), (5), (6), (7), (8), (9) \tag{11}$$

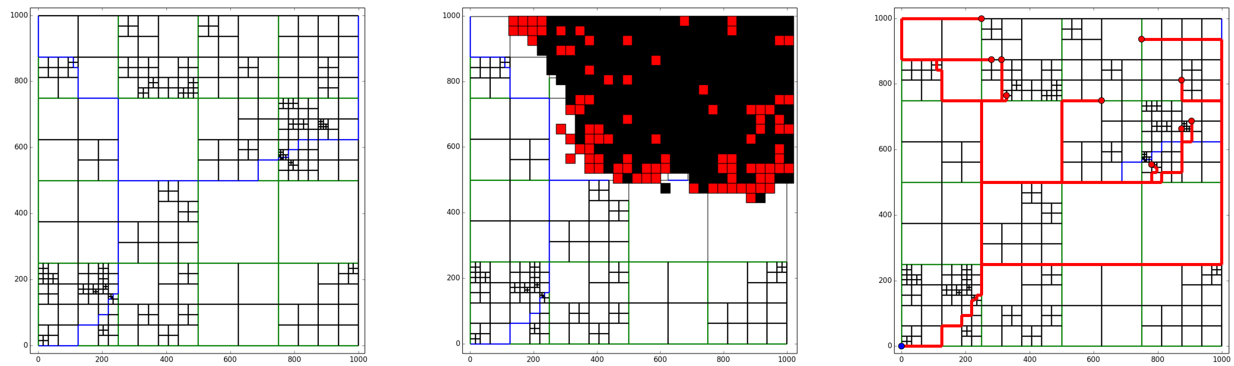
3 Realistic data instance generator

Catastrophic wildfire requiring large population evacuation are, thankfully, rare events. However, it means that obtaining useful data is difficult, and indeed this a key problem within the GEO-SAFE project. A significant part of the project revolves around simulation tools such as EXODUS [7], however, even simulated data was hard to come by.

Therefore, we opted for taking advantage of the project environment to contribute to this effort by generating our own "realistic" dataset. On the one hand, this approach may introduce biases since we must use models to generate realistic road networks and simulate wildfires. On the other hand, we believe that it will make it much more convenient for benchmarking algorithms in the future. As it turns out, the generated instance are challenging even though relatively modest in size, thus being interesting from an academic viewpoint as well.

3.1 Generation of road networks

The first step is to generate a graph standing for the road network. To this end, we used the *quadtree* model described in [3]. In a nutshell, this model starts with a single square formed by four nodes and four edges. At each iteration, a square is chosen and five nodes are added, one in the center of the square, and one on each edge connected by a perpendicular edge to the center node. A parameter r controls the *sprawl*, that is, the preference for splitting larger squares ($r < 1$) or smaller square ($r > 1$). The graphs generated in this manner share a many features with real road networks: they are planar, embedded in an Euclidean plane, have similar density distributions, path lengths are within a constant factor of the Euclidean distance, and the number of turns is logarithmic with high probability. An example of random quadtree network is shown in Figure 1a. The colors on the edges correspond to road capacity. To allocate capacities, we first compute a minimum Steiner tree spanning three randomly chosen nodes in high density areas ("cities") and connect these cities to the nearest corner of the outer square. The corresponding set of edges are given the highest capacity and are coloured in blue in Figure 1a. A second set of edges, forming a grid are given an intermediate capacity, they are coloured in green.



(a) road network (b) simulated wildfire (c) evacuation plan

Figure 1: An example of generated instance

3.2 Simulating wildfire

The second step consists in determining *safety due dates* for every edge of the evacuation tree, that is, a time after which the edge become unsafe. To this purpose we use a relatively simple fire propagation model. We chose to use a simple model based on two parameters: a constant *intensity* γ representing the type of fuel material as well as the temperature, and a *wind direction*. Indeed, the goal is not so much to accurately predict fire propagation, as it is to generate safety due dates consistent with a wildfire. Of course, should the authorities use this type of planning tools during a real event, then correctly predicting fire propagation would be among the most important factor.

The land area is discretized into squares of fixed size (we use another parameter to control this size) which can be in three states: *untouched*, *burning* and *burned*. The fire starts as a single burning square, then at each iteration, any untouched square adjacent of a burning square catches fire with probability $\gamma(\frac{\pi-A}{\pi})^2$, where γ stands for the intensity of the fire, and A is the angle between the wind an a vector going from the center of burning square to the center of the untouched square. Moreover, any burning square that did not propagate stop burning with probability γ^2. Figure 1b illustrate the state of the simulated wildfire, with burning squares in red and burned squares in black.

3.3 Generating evacuation plans

The third step consists in generating the actual evacuation plan, that is, an embedded tree connecting a set of evacuation nodes $\mathcal{E}$ to a safe node r. Here again, the goal is not to compute the best evacuation plans, however they must be representative of what would be actual plans.

We first randomly pick a predefined number of evacuation nodes among the nodes of the graph that are in the state *burned* or *burning* of the simulated fire. Then we use the convention that the safe zone is the furthest corner from the center of the fire. The evacuation tree is computed simply by using a shortest paths algorithm, however with respect to an arc labeling taking into account first the safety due date of the arc, and only then its length and its capacity.

At this point we have all the information we need to define a fire evacuation problem as defined in Section 2.3.

The tools we developed as well as the benchmarks instances we used in this paper can be accessed here: `https://github.com/ehebrard/evacsim`.

4 Formulations and heuristic

In this section, we propose two formulations for the problem: a 0–1 integer linear programing formulation and a constraint programming formulation. Then, we describe a simple compression heuristic able to find quickly an initial solution.

4.1 0–1 linear programming formulation

Let H denote an upper bound on the latest evacuation completion time on the evacuation nodes (the time by which the last evacuee leaves the evacuation node). We propose an integer 0-1 linear formulation that makes a discrete approximation of the problem. The set of discrete evacuation possible start times is equal to $\mathcal{H}_k = \{0, \ldots, H - \underline{p}_k\}$ where $\underline{p}_k = \lceil \frac{d_k}{u_k} \rceil$ is the smallest possible integer evacuation processing time. For an evacuation start time $t \in \mathcal{H}_k$, the set of possible evacuation processing times is $\mathcal{P}_{kt} = \{\underline{p}_k, \ldots, \min(d_k, H - t)\}$. Let $p_{mk} \in \mathcal{P}_{kt}$ denote the m^{th} smallest possible processing time for $m = 1, \ldots, |\mathcal{P}_{kt}|$. We introduce a 0–1 variable x_{ktm} equal to 1 if and only if $\delta_k = t$, $p_k = p_{mk}$ et $\lambda_k = \frac{d_k}{p_{mk}}$. By analogy to multi-mode scheduling problems, set $\{1, \ldots, |\mathcal{P}_{kt}|\}$ represent the set of processing modes available for scheduling an evacuation task that starts at time t. For a given mode m and a given time t, we denote by $I^i_{ktm} \subseteq \mathcal{H}_k$ the maximal discrete time interval such that

$$\forall \tau \in I^i_{ktm}, t \in [\tau + \sum_{q=1}^{i-1} t^q_k, \ldots, \tau + \sum_{q=1}^{i-1} t^q_k + p_{mk} - 1]$$

i.e. the set of evacuation start times τ in mode m that make evacuation on edge e^i_k in process at time t. This interval is precisely:

$$I^i_{ktm} = \{t - \sum_{q=1}^{i-1} t^q_k, \ldots, t - \sum_{q=1}^{i-1} t^q_k - p_{mk} + 1\} \cap \mathcal{H}_k$$

Given these elements , the problem can be expressed as the following 0–1 integer linear program:

$$\max W_{\min} \tag{12}$$
$$\text{s.t.}$$
$$W_{\min} \leq d_k(b_{e^i_k} - \sum_{t \in \mathcal{H}_k} \sum_{m=1}^{|\mathcal{P}_{kt}|} (t + p_{mk}) x_{ktm} - \sum_{q=1}^{i-1} t^q_k) \qquad \forall k \in \mathcal{E}, \forall i = 1, \ldots, |\mathcal{A}_k| \tag{13}$$
$$\sum_{t \in \mathcal{H}_k} \sum_{m=1}^{|\mathcal{P}_{kt}|} x_{ktm} = 1 \qquad \forall k \in \mathcal{E} \tag{14}$$
$$\sum_{J^i_k \in \mathcal{J}_e} \sum_{m=1}^{|\mathcal{P}_{k0}|} \sum_{\tau \in I^i_{ktm}} \lambda_{km} x_{ktm} \leq u_e \qquad \forall e \in \mathcal{A}, \forall t = 0, \ldots, H - 1 \tag{15}$$
$$x_{ktm} \in \{0, 1\} \qquad k \in \mathcal{E}, t \in \mathcal{H}_k, m \in \mathcal{P}_{kt} \tag{16}$$

4.2 Constraint Programming formulation

In [6], the NEPP was modeled using standard `cumulative` constraints. We adapt here this model for our problem. Let $\overline{x}$ (resp. $\underline{x}$) denote the largest (resp. smallest) value in the domain of a variable x. Given a set of tasks J with start time variable $s_i \in [\underline{s}_i, \overline{s}_i]$, processing time variable $p_i \in [\underline{p}_i, \overline{p}_i]$, height variable $\lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i]$ and a resource r of constant capacity u_r, recall that $\texttt{cumulative}((s_i, p_i, \lambda_i)_{i \in J}, u_r)$ enforces the relations

$$\sum_{i \in J | s_i \leq t < s_i + p_i} \lambda_i \leq u_r \quad \forall t \in \mathcal{H}$$

Consequently, to model the problem, it suffices to associate a task J_k^i to each arc on the evacuation path of each evacuation node $k \in \mathcal{E}$, with height variable $\lambda_k \in [1, u_k]$, start time variable $S_k^i \in [0, H - \frac{d_v}{u_v}]$, duration variable $p_k \in [\frac{d_v}{u_v}, d_v]$. A resource is defined per arc $e \in \mathcal{A}$, with capacity u_e.

The baseline constraint program for the evacuation planning problem is obtained by replacing constraints (6) in the problem formulation of Section 2.3 by:

$$\texttt{cumulative}((S_k^i, p_k, \lambda_k)_{J_k^i \in \mathcal{J}_e}, u_e) \quad \forall e \in \mathcal{A} \tag{17}$$

4.3 Heuristic

We propose a simple compression heuristic to find an initial upper bound. The heuristic is based on the assumption that scheduling all evacuation tasks at time 0 with the minimum evacuation rate yields a feasible solution, with a high cost. Starting from this solution ($\forall k \in \mathcal{E}$, we set the start time $s_k := 0$, the end time $e_k := d_k$, and the rate $\lambda_k := 1$). Now an iterative process starts where, at each iteration, the critical evacuation tasks, i.e. the one that minimizes the cost on some edge, is identified. Then, its duration is decreased and its height is consequently increased until (i) either no more height increase/duration decrease can be performed without exceeding an edge capacity or (ii) the task is not critical anymore and another task becomes critical. If case (i) occurs the process stops, otherwise, if case (ii) occurs, the compression process restarts with the new critical task unless the objective increase is smaller than a predefined parameter ϵ, in which case the process also stops. Due to the possibility of only left shifting a task k by $\frac{\epsilon}{d_k}$ at each iteration, this descent heuristic is of pseudo polynomial computational complexity.

5 Computational experiments

We generated 240 benchmark instances following the protocol described in Section 3. They are organized into three types of road networks: `Dense`, `Medium` and `Sparse` where the density refers to the number of intersections (respectively 400, 800 and 1200) in the land area. Notice that the graph has always 4 edges per node, so this corresponds to graph size. The impact on the instance is that larger graphs allow for more choices for the shortest paths and therefore longer independent paths. For every type of road network, we generated 4 classes of instances, with respectively 10, 15, 20 and 25 evacuation nodes. Finally, for every class we simulated 20 random wildfires and the subsequent evacuation trees.

We used CPLEX 12.7 to solve the MILP formulation with default settings and CPOptimizer 12.7 for solving the CP formulation. The heuristic was used to provide an initial solution to both solvers. We ran every method on every instance of the dataset with a time limit of 45 minutes on 4 cluster nodes, each with 35 Intel Xeon CPU E5-2695 v4 2.10GHz cores running Linux Ubuntu 16.04.4.

We provide here e few implementation details. We only used discrete evacuation rates as it was simpler to implement in the CP solver. It follows that Constraints (4) was implemented as an inequality ($\geq$). The number of edges on which to check the cumulative constraint was reduced thank to the observation that in a path on an evacuation only one edge is a bottleneck. It follows that only one edge per path has to be considered as a limited resource. Last, the opposite of the weighted slack was actually minimized, which amounts to a maximum weighted lateness objective. The results are displayed in table 1, where we give for each solver the average upper bound on each instance family and the optimality ratio, i.e. the percentage of verified optimal solutions found.

		CPO		MIP	
		ub	opt	ub	opt
dense_10	(20)	23294	0.95	58746	0.70
dense_15	(20)	161100	0.70	346404	0.00
dense_20	(20)	311676	0.45	659395	0.00
dense_25	(20)	531016	0.00	1044248	0.00
medium_10	(20)	48591	1.00	103151	0.55
medium_15	(20)	124921	0.70	306552	0.05
medium_20	(20)	276101	0.25	581273	0.00
medium_25	(20)	488282	0.10	955786	0.00
sparse_10	(20)	3747	1.00	14359	0.70
sparse_15	(20)	120173	0.65	288432	0.05
sparse_20	(20)	235771	0.35	554373	0.00
sparse_25	(20)	437618	0.00	868438	0.00
avg	(240)	230191	0.51	481763	0.17

Table 1: CPO vs MIP: upper bound and optimality ratio

We can first remark that the generated instances are hard to solve optimally for both solvers: on average only 51% of the instances are solved to optimality with the same behavior for the families: almost all the instances with10 evacuation node and a large part of the instances with 15 evacuation nodes car be solved to optimality, while instances with 20 evacuation nodes becomes much harder and the ones with 25 activities are intractable. As an outcome, our generator is able to produce computationally challenging instances.

In terms of comparison between integer and constraint programming, the integer program is significantly outperformed by CP, both in terms of optimality ratio and of upper bounds on maximum weighted lateness. This is both due to slower convergence time and memory issued due to the huge size of the IP model. As a typical example, the instance `medium_10_30_3_2` has 328147 binary variables and 3198 constraints after CPLEX preprocessing.

In terms of the obtained objective function values, on the 240 instances only 45 have negative values, meaning that in a majority of instances the evacuation could not be performed on some edge before the expected deadline. Interestingly, all these 45 instances were solved to optimality, which represents 36% of the 123 instances solved to optimality. 74% of the remaining instances, correspond to pessimistic scenarios where the evacuation road network is unable to ensure the evacuation of the the whole population before the traversal of some route segment would become critical. If such situation occurred in actual road network this could give helpful support to the authorities for increasing the capacity of specific road segments or to build better prescribed evacuation routes.

6 Concluding remarks

We have proposed a data instance generator and optimization frameworks for a computationally challenging evacuation planning problem, with an objective function tailored to the event of wildfire. This generator could be improved by incorporating more sophisticated fire propagation models and actual road networks. The generation of evacuation routes is also an optimization problem in itself. Feedback from the evacuation planning, in the case, which often occurred in our experiments, where obtained safety margin are not sufficient should be used to modified the evacuation routes accordingly. In terms of the evacuation planning problem we have proposed new integer and constraint programming formulation. To obtain competitive results with IP, one should obviously consider decomposition approach as the problem is huge. Continuous models could also be designed to reduce the number of variables. Even if CP obtains much better results, the vast majority of medium size instances could not be solved to optimality. As future research directions, we will specific global constraint that better capture the structure of the problem as well as dedicated search strategies. Finally, we believe that coupling our approach with simulation and/or stochastic–robust optimization will lead to useful decision support tools in case of response to wild fires.

Acknowledgement

This work is partially funded by the H2020-MSCA-RISE-2015 European project GEO-SAFE (id 691161)

References

[1] Hal E. Anderson. Aids to determining fuel models for estimating fire behavior. Technical Report 122, United States Department of Agriculture- Forest Service, Intermountain Forest and Range Experiment Station Ogden, UT 84401, April 1982.

[2] Vedat Bayram. Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science*, 2016. DOI: 10.1016/j.sorms.2016.11.001.

[3] David Eisenstat. Random road networks: the quadtree model. *CoRR*, abs/1008.4916, 2010.

[4] Caroline Even, Victor Pillac, and Pascal Van Hentenryck. NICTA evacuation planner: Actionable evacuation plans with contraflows. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 1143–1148, 2014.

[5] Caroline Even, Victor Pillac, and Pascal Van Hentenryck. Convergent plans for large-scale evacuations. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1121–1127, 2015.

[6] Caroline Even, Andreas Schutt, and Pascal Van Hentenryck. A constraint programming approach for non-preemptive evacuation scheduling. In *Principles and Practice of Con-*

straint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings, pages 574–591, 2015.

[7] Edward R. Galea, Mathew Owen, and Peter J. Lawrence. The EXODUS Model. *Fire Engineers Journal*, pages 26–30, 1996.

[8] Geo-safe - geospatial based environment for optimisation systems addressing fire emergencies, MSCA-RISE-2015 - Marie Skłodowska-Curie Research and Innovation Staff Exchange (RISE) European projet – id 691161. `http://fseg.gre.ac.uk/fire/geo-safe.html`. Accessed: June 16, 2018.

[9] Kanal Kumar, Julia Romanski, and Pascal Van Hentenryck. Optimizing infrastructure enhancements for evacuation planning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3864–3870, 2016.

[10] Victor Pillac, Manuel Cebrián, and Pascal Van Hentenryck. A column-generation approach for joint mobilization and evacuation planning. *Constraints*, 20(3):285–303, 2015.

[11] Victor Pillac, Caroline Even, and Pascal Van Hentenryck. A conflict-based path-generation heuristic for evacuation planning. *Transportation research part B*, (83):136–150, 2016.

[12] Victor Pillac, Pascal Van Hentenryck, and Caroline Even. A path-generation matheuristic for large scale evacuation planning. In *Hybrid Metaheuristics - 9th International Workshop, HM 2014, Hamburg, Germany, June 11-13, 2014. Proceedings*, pages 71–84, 2014.

[13] Julia Romanski and Pascal Van Hentenryck. Benders decomposition for large-scale prescriptive evacuations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3894–3900, 2016.

[14] Richard C. Rothermel. A mathematical model for fire spread predictions in wildland fuels. Technical Report 115, USDA For. Serv., lntermt. For. and Range Exp. Stn., Ogden, Utah, USA, 1972.

[15] Shahrooz Shahparvari, Prem Chhetri, Babak Abbasi, and Ahmad Abareshi. Enhancing emergency evacuation response of late evacuees: Revisiting the case of australian black saturday bushfire. *Transportation Research Part E: Logistics and Transportation Review*, 93:148 – 176, 2016.

[16] Alexander Stepanov and James MacGregor Smith. Modeling wildfire propagation with delaunay triangulation and shortest path algorithms. *European Journal of Operational Research*, 218(3):775 – 788, 2012.

[17] Anand Veeraswamy, Edwin R Galea, Lazaros Filippidis, Peter J Lawrence, and Robert J Gazzard. The simulation of urban-scale evacuation scenarios: Swinley forest fire. In *Proceedings 6th Int Symp Human Behaviour in Fire*, pages 221–232, 2015.

[18] David R. Weise and Gregory S. Biging. A qualitative comparison of fire spread models incorporating wind and slope effects. *Forest Science*, 43(2):170–180, 1997.