# A Conflict-Based Path-Generation Heuristic for Evacuation Planning

Victor Pillac*[1], Pascal Van Hentenryck[1,2], and Caroline Even[1,3]

[1]*NICTA, Victoria Research Laboratory*
Melbourne, Australia
[2]*Department of Computing and Information Systems, University of Melbourne*
Melbourne, Australia
[3]*Ecole des Mines de Nantes*
Nantes, France

September 2013

**Abstract**

Evacuation planning and scheduling is a critical aspect of disaster management and national security applications. This paper proposes a conflict-based path-generation approach for evacuation planning. Its key idea is to generate evacuation routes lazily for evacuated areas and to optimize the evacuation over these routes in a master problem. Each new path is generated to remedy conflicts in the evacuation and adds new columns and a new row in the master problem. The algorithm is applied to massive flood scenarios in the Hawkesbury-Nepean river (West Sydney, Australia) which require evacuating in the order of 70,000 persons. The proposed approach reduces the number of variables from 4,500,000 in a Mixed Integer Programming (MIP) formulation to 30,000 in the case study. With this approach, realistic evacuations scenarios can be solved near-optimally in real time, supporting both evacuation planning in strategic, tactical, and operational environments.

**Keywords:** Evacuation planning and scheduling, regional evacuation, disaster management, conflict-based path generation

*Corresponding author: victor.pillac@nicta.com.au

# 1 Introduction

Natural disasters, such as hurricanes, floods, and bushfires, affect numerous populated areas and may endanger the lives and welfare of entire populations. Evacuation orders are some of the most important decisions performed by emergency services: They ensure the safety of persons at risk by instructing them to evacuate the threatened region, be it a building (e.g., fire), a neighborhood (e.g., industrial hazard), or a whole region (e.g., flood). Evacuation planning also arises at strategic, tactical, and operational levels. At a strategic level, the goal is to design evacuation plans for specific areas and possible threats (e.g., evacuation plans for the surroundings of a nuclear power plant). At a tactical level, the goal is to design evacuation plans for an area facing an incoming threat (e.g., evacuation of a flood plain following high precipitations). Finally, at the operational level, the goal is to schedule an evacuation, possibly adjusting the evacuation plan in real-time as the threat unfolds.

Existing work in evacuation planning typically consider free-flow models in which evacuees are dynamically routed in the network. However, free-flow models do not conform to existing evacuation methodologies in which evacuated nodes are assigned specific evacuation routes (see, for instance, [21]).

In contrast, this paper presents an evacuation algorithm that follows recommended evacuation methodologies: It generates evacuation routes for evacuated nodes and uses a lexicographic objective function that first maximizes the number of evacuees and then postpones the evacuation as much as possible. The algorithm can be used for strategic and tactical planning and is fast enough to operate in real-time conditions, even for large evacuations.

From a technical standpoint, the algorithm can be broadly characterized as a Conflict-Based Path-Generation Heuristic (CPG for short), which shares some characteristics with column generation approaches. As in column generation, we decompose the problem by considering separately the generation of evacuation paths (subproblem) and their selection (master problem). However, a challenge of our application is the spatio-temporal nature of the problem: one evacuation path corresponds to multiple paths in the spatio-temporal graph modeling the actual scheduling of the evacuation. Therefore, a single path introduces multiple columns in the master problem, which increases the complexity the pricing of a new path. To tackle this limitation, the path-generation subproblem aims at finding a path of least cost under constraints, where the edge costs are derived from the conflicts and congestion in the incumbent evacuation.

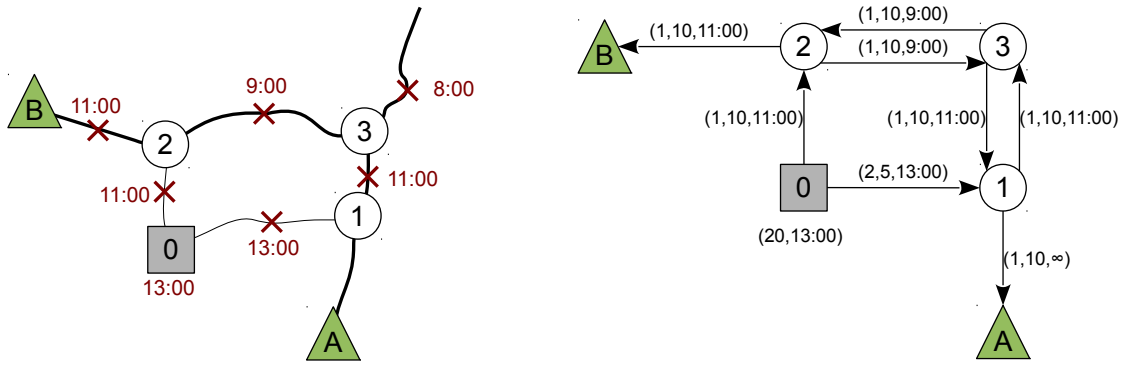
The CPG algorithm was evaluated on real-scale, massive flood scenarios in the Hawkesbury-Nepean river (West Sydney, Australia) which require evacuating in the order of 70,000 persons. Experimental results indicate that the CPG algorithm generates high-quality solutions in real time. On small instances, where optimal solutions can be found, the CPG algorithm finds optimal or near-optimal solutions. On real-scale instances, the results show that the CPG algorithm is capable of evacuating the entire Hawkesbury-Nepean region even if the population grows by 40%. The solution quality of the CPG algorithm can be bounded by free-flow models that provide (optimistic) upper bound on solution quality: The results indicate that its solutions (in terms of evacuees) are within 13% of free-flow models even if the population is increased by 200%. Finally, microscopic traffic simulations show that the solution produced by CPG are robust to different evacuees behaviours.

The remainder of this paper is organized as follows: Section 2 formulates the evacuation planning problem, Section 3 reviews related work, Section 4 presents three solution approaches, while Section 5 discusses two formulations to postpone the evacuation as much

as possible. Section 6 compares the performance of the proposed approaches on a set of realistic instances. Finally, Section 7 concludes this paper.

## 2 Problem Formulation

Figure 1 illustrates a general evacuation scenario. Fig. 1(a) presents an evacuation scenario with one evacuated node (0) and two safe nodes (A and B). In this example, the evacuated node 0 has to be evacuated by 13:00, considering that certain links become unavailable at different times (for instance, $(2,3)$ is cut at 9:00). This evacuation scenario can be represented as a graph $\mathcal{G} = (\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ where $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{S}$ are the set of evacuated, transit, and safe nodes respectively, and $\mathcal{A}$ is the set of edges. Each evacuated node $i$ is characterized by a number of evacuees $d_i$ and an evacuation deadline $\bar{f}_i$ (e.g., 20 and 13:00 for node 0 respectively), while each edge $e$ is associated with a triple $(s_e, u_e, \bar{f}_e)$, where $s_e$ is the travel time, $u_e$ is the capacity, and $\bar{f}_e$ is the time at which the edge becomes unavailable.




(a) Evacuation Scenario

(b) Evacuation Graph

Figure 1: Modeling of an Evacuation Planning Problem.

A common way to deal with the space-time aspects of evacuation problems is to discretize the planning horizon into time steps of identical length, and to work on a *time-expanded graph*, as illustrated in Fig. 2. This graph $\mathcal{G}^d = (\mathcal{N}^d = \mathcal{E}^d \cup \mathcal{T}^d \cup \mathcal{S}^d, \mathcal{A}^d)$ is constructed by duplicating each node from $\mathcal{N}$ for each time step. For each edge $(i,j) \in \mathcal{A}$ and for each time step $t$ in which edge $(i,j)$ is available, an edge $(i_t, j_{t+s_{(i,j)}})$ is created modeling the transfer of evacuees from node $i$ at time $t$ to node $j$ at time $t + s_{(i,j)}$. In addition, edges with infinite capacity are added to model the evacuees waiting at evacuated and safe nodes. Finally, all evacuated nodes (resp. safe nodes) are connected to a virtual super-source $v_s$ (super-sink $v_t$), modeling the inflow (outflow) of evacuees. Note that some nodes may not be connected to either the super-source or super-sink (in light gray in this example), and can therefore be removed to reduce the graph size. The problem is then to find a flow from $v_s$ to $v_t$ that models the movements of evacuees in space and time.

In this study, we consider a single threat scenario from which we derive the time when each evacuated node must be evacuated, and the time at which edges are closed. In addition, we ignore the dynamics of the actual evacuees movements, and assume that the edge capacity is fixed and does not depend on the flow. This is a necessary simplification that is compensated by the fact that edge capacities are set to ensure non-congested flow. In addition, we follow

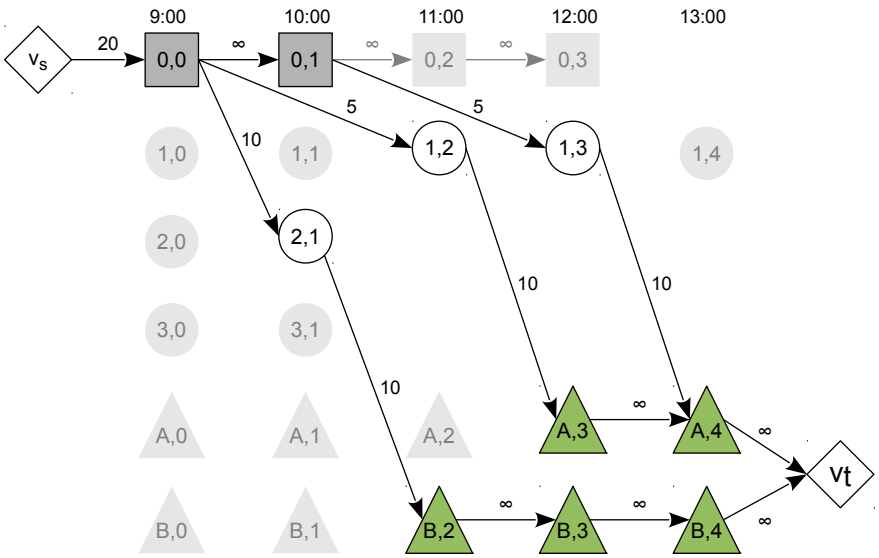



Figure 2: Time-Expanded Graph for the Evacuation Scenario With 1-hour Time Steps.

the practice in the field of emergency services operations and assume that all evacuees from a same node will be evacuated to the same safe node following a single path.

The objective is to first ensure that all evacuees reach a safe node, and then to delay the evacuation as much as possible. This second objective is motivated by the type of threat considered: we assume that evacuees safety is only threatened after the evacuation deadline. Therefore, it is of practical interest to evacuate them as late as possible, as this leaves more time to potentially refine the threat scenario and hence avoids unnecessary evacuations. The decisions that need to be made for each evacuated are the following: which safe node to evacuate to, which path to follow to reach the selected safe node, and how to schedule the departures over the horizon. Finally, the global evacuation plan and schedule must respect the capacity of the network, and ensure that no evacuee travels on an edge that has been cut.

# 3 Related work

According to Hamacher and Tjandra [10], evacuation planning can be tackled using either *microscopic* or *macroscopic* approaches. Microscopic approaches focus on modeling and simulating the evacuees individual behaviors, movements, and interactions. Macroscopic approaches, such as the three presented in this study, aggregate evacuees and model their movements as a flow in the evacuation graph.

To the best of our knowledge, all models are free-flow. A significant number of contributions attempt to solve flow problems directly derived from the time-expanded graph. For instance, Lu et al. [16, 15] propose three heuristics to design an evacuation plan with multiple evacuation routes per evacuated node, minimizing the time of the last evacuation. The authors show that in the best case the proposed heuristic is able to solve randomly generated instances of up to 50,000 nodes and 150,000 edges in under 6 minutes. Liu et al. [14] propose a Heuristic Algorithm for Staged Traffic Evacuation (HASTE), a similar algorithm that generates evacuation routes and schedule the evacuation of evacuated nodes in sequence.

The main difference between HASTE and the previous algorithms is that it relies on a Cell Transmission Model (CTM)[8] to model more accurately the flow of evacuees.

Acknowledging that all evacuated nodes may not be under the same level of threat, Lim et al. [12] consider a short-notice regional evacuation maximizing the number of evacuees reaching safety weighted by the severity of the threat. The authors propose two solution approaches to solve the problem, and present computational experiments on instances derived from the Houston-Galveston region (USA) with up to 66 nodes, 187 edges, and a horizon of 192 time steps.

Other authors have focused on modeling more accurately the transportation network. For example, Bretschneider and Kimms [5, 6] present a free-flow mathematical model that describes in detail the street network and, in particular, the lane configuration at intersections of the network. They present computational experiments on generated instances with a grid topology of up to 240 nodes, 330 edges, and considering 150 times steps. Bish and Sherali [4] present a model based on a CTM that assigns a single evacuation path to each evacuated node. Computational results include instances with up to 13 evacuated nodes, 2 safe nodes, and 72 edges.

Finally, dynamic aspects of evacuation have also been considered. For instance, Lin et al. [13] present a time expanded graph in which they allow for time-dependent attributes such as varying capacity or demand. The authors apply their findings on a case study considering the evacuation of a 11-floor building with approximately 60 nodes, 100 edges, and 60 time steps.

Microscopic approaches include the work by Richter et al. [20] who challenge two assumptions generally made: The existence of a central planning entity with global knowledge, and the ability of this entity to communicate order to evacuees. They propose a decentralized decision making approach supported by smartphones and mobile applications. We note however that our target applications, such as evacuations for floods and hurricanes, use central decision making and have the time and ability to communicate their decisions.

Column generation is an optimization technique which consists in considering only a subset of columns in a master problem and then iteratively generating columns of negative reduced cost (assuming minimization) by solving a pricing subproblem. It has been widely used to solve large-scale MIP problems, and we refer the interested reader to the book by Desaulniers et al. [9] and the study by Lübbecke and Desrosiers [17] for a recent review of techniques and applications of column generation. In particular, it has been used to solve multi-commodity network flow problems (MCNF) [1], integer MCNF [3], origin-destination MCNF [2], and MCNF with side constraints on paths [11]. However, a distinctive feature of evacuation planning is the dependency between paths in the time-expanded network. More precisely, a commodity (i.e., evacuees from a specific evacuated node) can only follow paths that correspond to the same physical path (sequence of edges in the evacuation graph). Therefore classical MCNF approaches cannot be applied directly, as one path in the evacuation model introduces multiple variables in the master problem. In addition, it is worth noting that heuristic column generation have mainly focused on solving the pricing subproblem heuristically. In contrast, our approach does not consider the pricing problem explicitly, but heuristically generates new paths. Similar ideas were also used by Coffrin et al. [7], and, to a lesser extent, by Massen et al. [18].

# 4 Proposed Approaches

In this section we present three approaches of increasing practical relevance to the decision maker. The first approach is based on free-flow models and is used as a bound to evaluate the other two models. The second approach uses a Mixed Integer Programming (MIP) formulation to ensure that evacuees from a same evacuated node follow a single evacuation path. Finally, the third approach is our conflict-based heuristic path generation algorithm.

## 4.1 Free-Flow Model

The Free-Flow model (FF) assumes that evacuees can flow freely in the graph. From a practical perspective, this corresponds to an ideal case in which the evacuees are given the order to evacuate and are then dynamically routed in the graph. In other words, evacuees do not know in advance the path to follow in the graph. Although of limited practical importance, this model provides a bound for more advanced evacuation planning.

$$\max \quad \Phi = \sum_{e \in \delta^-(v_t)} \varphi_e \tag{1}$$

$$\text{s.t.} \quad \sum_{e \in \delta^-(i)} \varphi_e - \sum_{e \in \delta^+(i)} \varphi_e = 0 \qquad \forall i \in \mathcal{N}^d \setminus \{v_s, v_t\} \tag{2}$$

$$\varphi_e \leq u_e \qquad \forall e \in \mathcal{A}^d \tag{3}$$

$$\varphi_e \geq 0 \qquad \forall e \in \mathcal{A}^d \tag{4}$$

Figure 3: The Free-Flow model (FF).

Figure 3 presents the Free Flow model (FF). $\varphi_e$ is the flow of evacuees on edge $e \in \mathcal{A}^d$, and $\delta^-(i)$ ($\delta^+(i)$) is the set of inbound (outbound) edges adjacent to node $i$. The objective (1) is to maximize the number of evacuees reaching a safe node. Constraints (2) ensure the conservation of the flow in the graph, while constraints (3) enforce the capacity on edges. The demand of the evacuations nodes is modeled implicitly as a capacity on the edges connecting them to the super-source ($u_{(v_s,i)} = d_i$). Note that the flow of evacuees is considered as a continuous quantity. This is motivated by the fact that the considered number of evacuees and edge capacity are already approximations of the reality, thus a unitary granularity is not necessary.

## 4.2 Restricted-Flow Model

The restricted-Flow (RF) model enforces the constraint that each node is evacuated along a single path. They can be thought of as a form of multi-commodity flows but the spatio-temporal nature of evacuation planning introduces some key differences discussed later in the paper. The formulation of the model is interesting in that it expresses some of its constraints using the *evacuation graph* and others using the *time-expanded graph*. In general, the evacuation graph is used to enforce constraints on paths, while the time-expanded graph is instrumental in stating the flow constraints. Obviously, both graphs are necessary for some

constraints. For simplicity, we denote $e_0$ the edge in the evacuation graph that corresponds to edge $e$ in the time-expanded graph, projecting out the time information.

Figure 4 presents the RF model. It introduces a binary variable $x^k_{e_0}$ which is equal to 1 if and only if edge $e_0 \in \mathcal{A}$ belongs to the evacuation path for evacuated node $k$, and a continuous variable $\varphi^k_e$ equal to the flow of evacuees from evacuated node $k$ on edge $e \in \mathcal{A}^d$. The objective (5) is to maximize the number of evacuees reaching a safe node. Constraints (6) ensure that exactly one path is used to route the flow coming from a same evacuated node in the evacuation graph, while constraints (7) ensure the continuity of the path. Constraints (8) ensure the flow conservation through the time-expanded graph. Constraints (9) enforce the capacity of each edge in the time-expanded graph. Constraints (10) ensure that there is no flow of evacuees coming from an evacuated node $k$ if edge $e$ is not part of the evacuation path for $k$.

$$\max \quad \Phi = \sum_{e \in \delta^-(v_t)} \sum_{k \in \mathcal{E}} \varphi^k_e \tag{5}$$

$$\text{s.t.} \quad \sum_{e_0 \in \delta^+_0(k)} x^k_{e_0} = 1 \qquad \forall k \in \mathcal{E} \tag{6}$$

$$\sum_{e_0 \in \delta^-_0(i)} x^k_{e_0} - \sum_{e_0 \in \delta^+_0(i)} x^k_{e_0} = 0 \qquad \forall k \in \mathcal{E}, i \in \mathcal{T} \tag{7}$$

$$\sum_{e \in \delta^-(i)} \varphi^k_e - \sum_{e \in \delta^+(i)} \varphi^k_e = 0 \qquad \forall i \in \mathcal{N}^d \setminus \{v_s, v_t\}, k \in \mathcal{E} \tag{8}$$

$$\sum_{k \in \mathcal{E}} \varphi^k_e \leq u_e \qquad \forall e \in \mathcal{A}^d \tag{9}$$

$$\varphi^k_e \leq u_e * x^k_{e_0} \qquad \forall e \in \mathcal{A}^d, k \in \mathcal{E} \tag{10}$$

$$\varphi^k_e \geq 0 \qquad \forall e \in \mathcal{A}^d, k \in \mathcal{E} \tag{11}$$

$$x^k_e \in \{0, 1\} \qquad \forall e \in \mathcal{A}^d, k \in \mathcal{E} \tag{12}$$

Figure 4: The Restricted-Flow model (RF).

## 4.3 Conflict-Based Heuristic Path Generation

The main drawback of the RF model is its complexity, both in terms of number of variables and constraints. As the experimental results demonstrate, the model is computationally intractable even for small instances. To address this issue, we propose a conflict-based heuristic path generation approach (CPG) that separates the generation of evacuation paths from the scheduling of the evacuation.

Algorithm 1 gives an overview of the approach. First, an initial set of paths $\Omega'$ is generated (line 1) and a master problem is solved to find an evacuation schedule that maximizes the number of evacuees reaching safety (line 2). The procedure then identifies *critical* evacuated nodes (line 4), which are not fully evacuated, or evacuated early. This information is later used to generate new paths (line 5). Finally the scheduling problem is solved including the newly generated paths (line 6). The last four steps are repeated for a given number of iterations or

until a predefined number of non-improving iterations has been reached (line 3).

**Algorithm 1** The Conflict-Based Path Generation.

**HN-Input:** $\mathcal{G}$ the evacuation graph, $\mathcal{G}^d$ the time-expanded graph.
**Output:** $\mathcal{S}$ the best solution found

1: $\Omega' \leftarrow \texttt{generatePaths}\,(\mathcal{G}, \emptyset, \mathcal{E}, \emptyset)$ ▷ Subproblem
2: $\mathcal{S} \leftarrow \texttt{scheduleEvacuation}\left(\Omega', \mathcal{G}, \mathcal{G}^d\right)$ ▷ Master problem
3: **while** stopping criterion not met **do**
4: $\quad \mathcal{E}_c \leftarrow \texttt{findCriticalEvacuatedNodes}\,(\mathcal{S})$
5: $\quad \Omega' \leftarrow \Omega' \cup \texttt{generatePaths}\,(\mathcal{G}, \Omega', \mathcal{E}_c, \mathcal{S})$ ▷ Subproblem
6: $\quad \mathcal{S} \leftarrow \texttt{scheduleEvacuation}\left(\Omega', \mathcal{G}, \mathcal{G}^d\right)$ ▷ Master problem
7: **end while**
8: **return** $\mathcal{S}$

The master problem can be solved using a mixed integer program. Let $\Omega$ be the set of all feasible paths between evacuated nodes and safe nodes and $\Omega_k$ be the subset of evacuation paths for evacuated node $k$. We define a binary variable $x_p$ which takes the value of 1 if and only if path $p \in \Omega$ is selected, a continuous variable $\varphi_p^t$ representing the number of evacuees to start evacuating on path $p$ at time $t$, and a continuous variable $\varphi_k$ accounting for the number of non-evacuated evacuees in node $k$. In addition, we denote by $\omega(e)$ the subset of paths that contain edge $e$ and by $\tau_p^e$ the number of time steps required to reach edge $e$ when following path $p$. Finally, we note $\mathcal{H}_p \subseteq \mathcal{H}$ the subset of time steps in which path $p$ is usable, and $u_p$ the capacity of path $p$.

$$\max \quad \sum_{p\in\Omega} \sum_{t\in\mathcal{H}_p} \varphi_p^t \tag{13}$$

$$\text{s.t.} \quad \sum_{p\in\Omega_k} x_p = 1 \qquad \forall k \in \mathcal{E} \tag{14}$$

$$\sum_{p\in\Omega_k} \sum_{t\in\mathcal{H}_p} \varphi_p^t + \varphi_k = d_k \qquad \forall k \in \mathcal{E} \tag{15}$$

$$\sum_{\substack{p\in\omega(e) \\ t-\tau_p^e\in\mathcal{H}_p}} \varphi_p^{t-\tau_p^e} \le u_e \qquad \forall e \in \mathcal{A}, t \in \mathcal{H} \tag{16}$$

$$\sum_{t\in\mathcal{H}_p} \varphi_p^t \le |\mathcal{H}_p| x_p u_p \qquad \forall p \in \Omega \tag{17}$$

$$\varphi_p^t \ge 0 \qquad \forall p \in \Omega, t \in \mathcal{H}_p \tag{18}$$

$$\varphi_k \ge 0 \qquad \forall k \in \mathcal{E} \tag{19}$$

$$x_p \in \{0, 1\} \qquad \forall p \in \Omega \tag{20}$$

Figure 5: The evacuation scheduling problem (CPG-MP).

Figure 5 presents the evacuation scheduling problem CPG-MP. The objective (13) maximizes the total flow of evacuees, which is equivalent to the number of evacuees reaching safety. Constraints (14) ensure that exactly one path is selected for each evacuated node, while con-

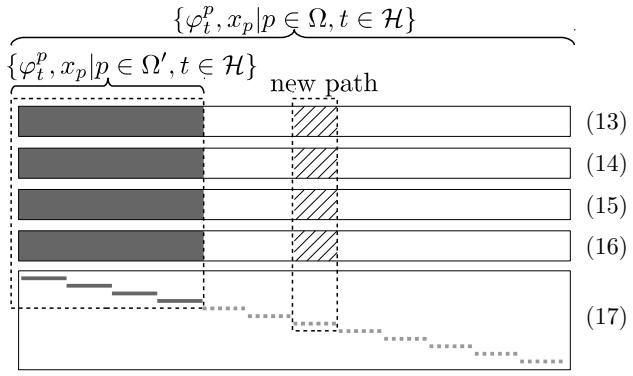



Figure 6: The Structure of the Evacuation Scheduling Master Problem Matrix.

straints (15) account for the number of evacuated and non-evacuated evacuees. Constraints (16) enforce the capacity on the edges of the graph. Finally, constraints (17) ensures that there is no flow on paths that are not selected. It is interesting to observe that the master model does not use a variable for each edge $e$ and time step $t$. Instead, it reasons in terms of variables $\varphi_p^t$ which indicate how many evacuees leave along path $p$ at time $t$.

In practice, we only consider a subset of evacuation paths $\Omega' \subseteq \Omega$ each time we solve CPG-MP. Fig. 6 depicts the structure of the master problem matrix. Horizontal blocks represent groups of constraints numbered as in Figure 5, while the shaded areas represent non-null coefficients in the matrix. Note that each constraint in group (17) only involves variables associated with the corresponding path and must be dynamically added to the model whenever a new path is considered. Nonetheless, a solution of CPG-MP considering the subset of paths $\Omega'' \subset \Omega'$ is also a feasible when considering the set $\Omega'$. Hence the solution from the previous iteration is used as starting solution for the current iteration.

Traditionally, the generation of new columns searches for a column of positive reduced costs (assuming maximization). Because of the spatio-temporal nature of this application, and the fact that a path corresponds to multiple columns and introduces a new constraint, we follow a different approach. We use a conflict-based path generation which relies on problem-specific knowledge to generate new columns that will potentially improve the objective function of the master problem. First, we identify the subset $\mathcal{E}' \subseteq \mathcal{E}$ of critical evacuated nodes, i.e., nodes that are not fully evacuated in the current solution. Then, we include in $\mathcal{E}'$ all the evacuated nodes whose evacuation paths share at least one edge with a node from $\mathcal{E}'$. Finally, we generate new paths for the critical evacuated nodes $\mathcal{E}'$ by solving the following multiple-origins, multiple-destinations shortest path problem:

$$\min \quad \sum_{k \in \mathcal{E}'} \sum_{e \in \mathcal{A}} c_e y_e^k \tag{21}$$

$$\text{s.t.} \quad \sum_{e \in \delta^-(i)} y_e^k - \sum_{e \in \delta^+(i)} y_e^k = 0 \qquad \forall i \in \mathcal{T}, k \in \mathcal{E}' \tag{22}$$

$$\sum_{e \in \delta^+(k)} y_e^k = 1 \qquad \forall k \in \mathcal{E}' \tag{23}$$

$$y_e^k \in \{0, 1\} \qquad \forall k \in \mathcal{E}', e \in \mathcal{A} \tag{24}$$

where $y_e^k$ is a binary variable taking the value of 1 if and only if edge $e$ belongs to the path

generated for evacuated node $k$, and $c_e$ is the cost assigned to edge $e$. In order to generate diverse evacuation paths, the cost $c_e$ of edge $e$ is adjusted at each iteration using the following linear combination of the edge's travel time $s_e$, the number of occurrences of $e$ in the current set of paths, and the utilization of $e$ in the current solution:

$$c_e = \alpha_t \frac{s_e}{\max_{e \in \mathcal{E}} s_e} + \alpha_c \frac{\sum_{\substack{p \in \Omega' \\ e \in p}} 1}{|\Omega'|} + \alpha_u \frac{\sum_{\substack{p \in \Omega' \\ e \in p}} \sum_{t \in \mathcal{H}_p} \varphi_p^t}{u_e} \tag{25}$$

where $\alpha_t$, $\alpha_c$, and $\alpha_u$ are positive weights which sum is equal to 1.

# 5 Postponing the Evacuation

The three models presented in the previous section share the common objective of maximizing the number of evacuees reaching safety. However, in the practical applications considered, stakeholders are also interested in delaying the evacuation as much as possible. This is motivated by the fact that, as the disaster unfolds, more information is available on the nature, extent, and timing of the threat, for instance with more accurate weather forecasts. Consequently, in this section we propose two formulations to maximize the time of the first evacuation, and design evacuation plans that not only guarantee that all evacuees can reach safety, but are also of practical relevance.

The first approach consists in solving a model maximizing the number of evacuees reaching safety and then using a post-optimization to maximize the time of the first evacuation. We illustrate this approach with the FF model, leading to the FF-E formulation. Similar transformations are applied to the RF and CPG models, defining the RF-E and CCG-E formulations. Note that, in the case of CPG-E, new columns can be generated either when maximizing the number of evacuees or when maximizing the time of the first evacuation. Let $x_t$ be a binary variable that takes the value of 1 if at least one evacuee leaves any evacuated node at time step $t$. In addition, we define the continuous variable $\Delta$ as the time of the first evacuation. We define $\Delta_{\text{UB}}$ as the time at which the first evacuated node is flooded. The FF-E model is defined as follows:

$$\max \quad \Delta \tag{26}$$

$$\text{s.t.} \quad (2) - (4)$$

$$\sum_{e \in \delta^+(v_s)} \varphi_e \geq \Phi_{max} \tag{27}$$

$$\sum_{i \in \mathcal{E}_t^d} \sum_{e \in \delta^+(i)} \varphi_e \leq x_t \sum_{i \in \mathcal{E}_t^d} \sum_{e \in \delta^+(i)} u_e \qquad \forall t \in \mathcal{H} \tag{28}$$

$$\Delta \leq t x_t + (1 - x_t)\Delta_{\text{UB}} \qquad \forall t \in \{1, \ldots, \Delta_{\text{UB}}\} \tag{29}$$

$$\Delta \geq 0 \tag{30}$$

$$x_t \in \{0, 1\} \qquad \forall t \in \mathcal{H} \tag{31}$$

In this model, the objective (26) is to maximize the time of the first evacuation $\Delta$. Constraint (27) ensures that the flow is at least the max flow $\Phi_{max}$ found in the first step. Constraints (28) define the binary variables $x_t$, constraints (29) ensure that $\Delta$ is bounded by the time of the first evacuation.

The drawback of the explicit formulations is that they introduce a binary variable per time step and require to perform two optimizations, leading to increased running times. Therefore, we introduce an implicit method that maximizes the number of evacuees and the scheduling of the evacuation in a single step. With the purpose of modeling "non-evacuated" evacuees, we add an edge between the last time-copy of each evacuated node and the super-sink, with zero travel time and infinite capacity. The scheduling of the evacuation is achieved by associating a penalty with edges leaving an evacuated node inversely proportional to the time slice in which they belong. Formally, let $c_{(i,j)}$ the cost of edge $(i,j) \in \mathcal{A}^d$, defined as:

$$c_{(i,j)} = \begin{cases} c_{ne} & \text{if } i \in \mathcal{E}^d, j = v_t \\ \frac{H-t(i)}{H} & \text{if } i \in \mathcal{E}^d, j \in \mathcal{T}^d \\ 0 & \text{otherwise} \end{cases}$$

where $t(i)$ is the time slice of time-node $i$, and $c_{ne}$ is a high penalty for non-evacuated evacuees. The Implicit Free Flow model (FF-I) minimizes the total cost of the flow for all edges subject to the conservation of the flow as follows:

$$\min \quad \sum_{e \in \mathcal{A}^d} c_e \varphi_e \tag{32}$$

$$\text{s.t.} \quad (2) - (4)$$

$$\varphi_{(v_s,i)} \geq d_i \qquad \forall i \in \mathcal{E} \tag{33}$$

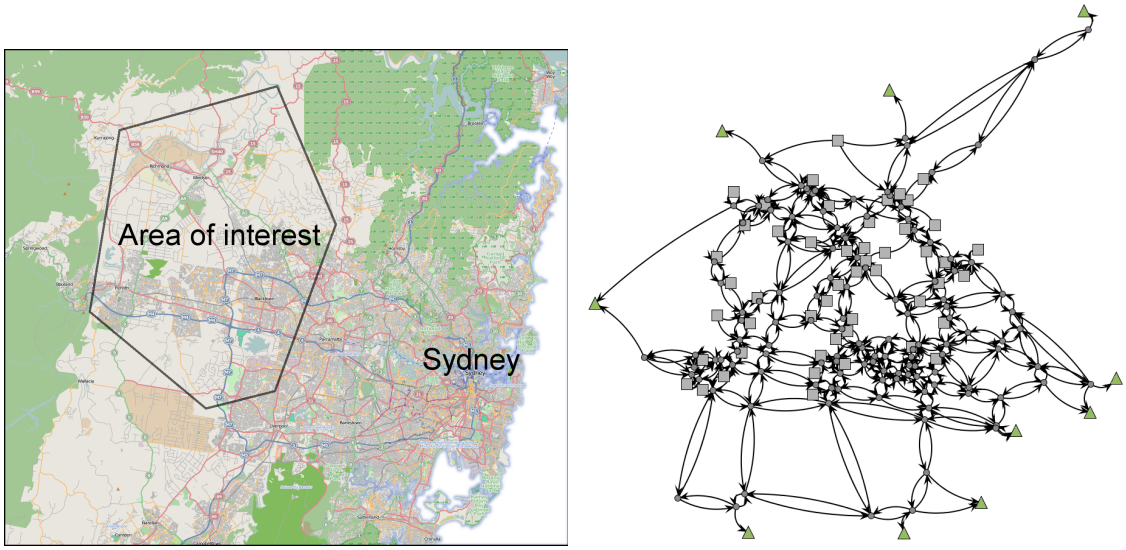
Similarly to the explicit formulations, this transformation is extended to the RF and CPG models, leading to the RF-I and CPG-I formulations.

# 6 Computational Experiments

## 6.1 Case Study

To assess the performance of our algorithms, we considered the evacuation of the Hawkesbury-Nepean (HN) floodplain, located North-West of Sydney (see Fig. 7(a)), for which a 1-in-200 years flood will require the evacuation of 70,000 persons. The resulting evacuation graph, illustrated in Fig. 7(b), contains 50 evacuated nodes, 10 safe nodes, 125 transit nodes, and 458 edges. We considered a horizon of 18 hours with a time step of 5 minutes (starting at 00h00). The evacuation deadlines and times at which edges are cut were derived from a flooding scenario similar to the historical 1867 flood [21].

In addition, we generated two sets of instances based on the case study. Instances HN-Rn were generated by first selecting the n $\in [2, 50]$ earliest flooded evacuated nodes, and then reducing the graph by retaining only the nodes and edges which are part of the shortest path between each evacuated node and the closest safe node. On the other hand, instances HN-Ix have the same evacuation graph as HN with a number of evacuees scaled by a factor of x $\in [1.1, 3.0]$. All approaches were implemented using Java 7 and Gurobi 5.1.1, and experiments were conducted on an Ubuntu 12.04 64bits machine with a 2.4Ghz 8-cores Intel Xeon processor and 32Gb of RAM. Results are an average over 10 runs given the randomized nature of parts of the algorithms and of Gurobi internal heuristics.




(a) Map of the area of interest (b) Evacuation graph for instance HN

Figure 7: Geographical location of the case study

## 6.2 Comparison of explicit and implicit formulations

Table 1 reports the CPU time and evacuation start time for each of the HN-Rn instances and the Free Flow (FF), Restricted Flow (RF), and Path Generation (CPG) approaches, using both Implicit (I) and Explicit (E) formulations. Note that all approaches were able to evacuate the totality of the evacuees in the reported solutions. These results highlight the practical limitation of the RF-E and RF-I approaches, which are not able to terminate in the enforced 30min time limit for instances with more than 5 evacuated nodes. Although the RF-I approach is able to find solutions for instances HN-R08 to HN-R30, this solution is of significantly lower quality than those of CPG-I. In addition, it appears that the CPG formulations produce solutions very similar to those produced by RF for the three smallest instances and competitive with FF for all instances, which means that CPG is able to find high-quality solutions with a reduced number of evacuation paths. Note that the reduced graph of instances HN-Rn implies that there exists only a limited number of paths departing each evacuated node, hence the similarities between the CPG and FF solutions. Finally, the table illustrates the benefits of the implicit formulation over the explicit one, as it significantly reduces the computational times while maintaining similar solution quality. Consequently, in the following, we focus on the implicit formulations.

## 6.3 Results on real-size instances

Table 2 presents computational results for the original HN instance and the HN-Ix instances for the three approaches and implicit formulations. The first column reports the number of paths generated, the second and third give the number of columns and rows in the MIP, the fourth reports CPU times, the sixth contains the percentage of evacuees reaching safety, and finally the seventh reports the time of the first evacuation. As expected, the RF-I approach is unable to find a feasible solution in the 30min time limit, while the CPG-I can solve all

| | | CPU Time (s) | | | Evacuation Start Time | | |
|---|---|---|---|---|---|---|---|
| **App.** | **Instance** | **E** | **I** | **Speedup** | **E** | **I** | **Gap** |
| **FF** | **HN-R02** | 1.0 | 0.0 | - | 13h35 | 13h35 | +0h00 |
| | **HN-R03** | 2.0 | 0.0 | - | 14h15 | 14h15 | +0h00 |
| | **HN-R05** | 0.0 | 0.0 | - | 12h25 | 12h25 | +0h00 |
| | **HN-R08** | 12.0 | 0.0 | - | 13h10 | 13h10 | +0h00 |
| | **HN-R10** | 11.0 | 0.0 | - | 13h10 | 13h10 | +0h00 |
| | **HN-R20** | 39.0 | 0.0 | - | 12h10 | 12h05 | +0h05 |
| | **HN-R30** | 28.0 | 0.0 | - | 10h05 | 10h00 | +0h05 |
| | **HN-R40** | 32.0 | 1.0 | 32.0 | 03h15 | 03h10 | +0h05 |
| | **HN-R50** | 28.0 | 1.0 | 28.0 | 03h15 | 03h10 | +0h05 |
| **RF** | **HN-R02** | 0.0 | 0.0 | - | 13h35 | 13h35 | +0h00 |
| | **HN-R03** | 18.0 | 1.0 | 18.0 | 14h00 | 14h00 | +0h00 |
| | **HN-R05** | 323.0 | 51.0 | 6.3 | 12h25 | 12h05 | +0h20 |
| | **HN-R08** | *1800.0* | *1800.0* | - | - | *07h55* | - |
| | **HN-R10** | *1800.0* | *1800.0* | - | - | *09h40* | - |
| | **HN-R20** | *1800.0* | *1800.0* | - | - | *00h00* | - |
| | **HN-R30** | *1800.0* | *1800.0* | - | - | *03h05* | - |
| | **HN-R40** | *1800.0* | *1800.0* | - | - | - | - |
| | **HN-R50** | *1800.0* | *1800.0* | - | - | - | - |
| **CPG** | **HN-R02** | 0.1 | 0.0 | - | 13h35 | 13h35 | +0h00 |
| | **HN-R03** | 0.9 | 0.2 | 4.7 | 13h55 | 13h55 | +0h00 |
| | **HN-R05** | 2.5 | 1.1 | 2.3 | 12h25 | 12h25 | +0h00 |
| | **HN-R08** | 7.7 | 1.1 | 6.7 | 12h10 | 11h53 | +0h17 |
| | **HN-R10** | 11.5 | 1.9 | 5.9 | 12h10 | 11h55 | +0h16 |
| | **HN-R20** | 13.4 | 10.1 | 1.3 | 11h12 | 11h35 | -0h23 |
| | **HN-R30** | 14.5 | 8.3 | 1.7 | 09h25 | 09h35 | -0h10 |
| | **HN-R40** | 9.7 | 2.6 | 3.8 | 03h15 | 03h11 | +0h05 |
| | **HN-R50** | 10.6 | 2.7 | 3.9 | 03h15 | 03h10 | +0h05 |

Table 1: Experimental Results on Reduced-Size Instances.

instances in under 30s. This table also highlights the dramatic reduction in model size that the CPG-I approach provides, with a number of columns reduced from 4.4 millions to 32 thousands, and a number of constraints reduced from 5.7 millions to 79 thousands when compared with the RF-I approach. Interestingly, the CPG-I model contains fewer variables that the simpler FF-I model. If we consider the percentage of evacuees to reach safety, we can note that the FF-I model is always able to evacuate 100% of evacuees, while the CPG-I approach finds optimal solutions for instances up to HN-I1.4 (i.e., with a population increased by 40% with respect too the current census). The CPG-I approach is still able to evacuate 87% of the population in scenarios where the population is increased threefold. Finally, CPG-I produces schedules that start much earlier than FF-I.

These results are particularly compelling considering that the free-flow models are not realistic and are only useful to provide upper bounds on solution quality. Contrary to the reduced instances, there are many paths that the free-flow models can exploit, making these models significantly over-optimistic. Although we cannot retrace exactly the paths followed by each evacuee, a closer look at instance HN shows that FF-I uses more than one evacuation path for at least 11 evacuated nodes. In addition, the flow of evacuees is split at 18 transit nodes (versus 0 for CPG-I), and FF-I uses 380 edges (82% of total) while CPG-I uses only

| App. | Instance | Num. Routes | Num. Cols ($10^3$) | Num. Rows ($10^3$) | CPU Time (s) | Perc. Evac. | Evac. Start |
|---|---|---|---|---|---|---|---|
| **FF-I** | **HN** | | 88 | 32 | 1 | 100% | 8h35 |
| | **HN-I1.1** | | 88 | 32 | 1 | 100% | 8h15 |
| | **HN-I1.2** | | 88 | 32 | 2 | 100% | 7h55 |
| | **HN-I1.4** | | 88 | 32 | 2 | 100% | 7h25 |
| | **HN-I1.7** | | 88 | 32 | 2 | 100% | 7h05 |
| | **HN-I2.0** | | 88 | 32 | 3 | 100% | 6h10 |
| | **HN-I2.5** | | 88 | 32 | 3 | 100% | 5h05 |
| | **HN-I3.0** | | 88 | 32 | 3 | 100% | 3h15 |
| **RF-I** | **HN** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I1.1** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I1.2** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I1.4** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I1.7** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I2.0** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I2.5** | | 4440 | 5655 | 1800 | - | - |
| | **HN-I3.0** | | 4440 | 5655 | 1800 | - | - |
| **CPG-I** | **HN** | 140 | 32 | 79 | 15 | 100% | 3h20 |
| | **HN-I1.1** | 140 | 32 | 79 | 23 | 100% | 2h25 |
| | **HN-I1.2** | 143 | 32 | 79 | 19 | 100% | 1h50 |
| | **HN-I1.4** | 143 | 32 | 79 | 25 | 100% | 0h20 |
| | **HN-I1.7** | 102 | 23 | 79 | 2 | 98% | 0h00 |
| | **HN-I2.0** | 100 | 23 | 79 | 2 | 92% | 0h00 |
| | **HN-I2.5** | 110 | 25 | 79 | 3 | 91% | 0h00 |
| | **HN-I3.0** | 116 | 26 | 79 | 8 | 87% | 0h00 |

Table 2: Experimental Results on Real-Size Instances.

169 (37%). Finally, applying FF-I to an instance reduced to the nodes and edges present in the CPG-I solution yields a evacuation starting at 3h40 (compared to 3h20 for CPG-I), with 2 evacuated nodes with more than one evacuation path, and a flow split on at least 7 transit nodes. This illustrates that the gap in performance between the two approaches is due to the fact that FF-I distributes the flow of evacuees from one zone over the entire evacuation graph, which is not realistic in practice.

## 6.4 Comparison with upper bound

Observing that the flow of evacuees is split in a number of transit nodes in FF, we derive a model, namely $\overline{\text{FF}}$, that prevents this behavior and provides an upper bound for the evacuation time. In this model, we define a binary variable for each edge of the static graph equal to one if and only if the edge is used. In addition, we add constraints that ensure that the number of used outbound edges is lower than the number of used inbound edges for each transit node. To prevent oscillating flows that will open as many edges as required, we add extra constraints that force each edge to be used in a single direction. Note that both additional constraints are sound from a practical perspective. The same constraints were also added to the CPG model for comparison, leading to the $\overline{\text{CPG}}$ approach.

Table 3 presents a comparison of $\overline{\text{FF}}$ and $\overline{\text{CPG}}$ for the HN and HN-Ix instances. Two observations can be made from the results. First, the new constraints have a dramatic impact

on the performance of $\overline{\text{FF}}$, which does not find the optimal solution in two hours, while they have a limited impact on $\overline{\text{CPG}}$ which still terminates under one minute on average. Second, the schedule of the evacuation produced by $\overline{\text{FF}}$ is much closer to $\overline{\text{CPG}}$ than FF is to CPG. This supports the claim that free flow models produce solutions that are overly optimistic when practical constraints are taken into account.

| App. | Instance | Num. Cols ($10^3$) | Num. Rows ($10^3$) | CPU Time (s) | MIP Gap. | Perc. Evac. | Evac. Start |
|---|---|---|---|---|---|---|---|
| $\overline{\text{FF}}$ | **HN** | 89 | 34 | 7200 | 4.15% | 100% | 3h40 |
| | **HN-I1.1** | 89 | 34 | 7200 | 4.74% | 100% | 2h50 |
| | **HN-I1.2** | 89 | 34 | 7200 | 4.09% | 100% | 2h40 |
| | **HN-I1.4** | 89 | 34 | 7200 | 3.87% | 100% | 5h05 |
| | **HN-I1.7** | 89 | 34 | 7200 | 2.90% | 100% | 3h00 |
| | **HN-I2.0** | 89 | 34 | 7200 | 2.58% | 100% | 1h15 |
| | **HN-I2.5** | 89 | 34 | 7200 | 3.77% | 100% | 0h00 |
| | **HN-I3.0** | 89 | 34 | 7200 | 63.54% | 96% | 0h00 |
| $\overline{\text{CPG}}$ | **HN** | 45 | 79 | 188 | - | 100% | 3h15 |
| | **HN-I1.1** | 20 | 79 | 4 | - | 100% | 3h25 |
| | **HN-I1.2** | 32 | 79 | 74 | - | 100% | 1h45 |
| | **HN-I1.4** | 23 | 79 | 10 | - | 100% | 1h20 |
| | **HN-I1.7** | 20 | 79 | 10 | - | 97% | 0h20 |
| | **HN-I2.0** | 20 | 79 | 11 | - | 92% | 0h00 |
| | **HN-I2.5** | 23 | 79 | 16 | - | 90% | 0h00 |
| | **HN-I3.0** | 29 | 79 | 24 | - | 84% | 0h00 |

Table 3: Experimental results for the $\overline{\text{FF}}$ and $\overline{\text{CPG}}$ models.

## 6.5 Validation through traffic simulation

The optimization approaches presented in this work assume that the evacuees (or vehicles) flow over the evacuation network in a continuous and aggregated manner. In the real world however, evacuees are independent agents that move along the edges and show different behaviors in response to the evacuation plan. To assess the fitness and robustness of the results from the optimization, we introduce an agent-based traffic simulation based on the MATSIM [19] simulation package.

In this simulation, each evacuee is modeled as an agent with an individual plan composed by a start location (its evacuated area), a final destination (the chosen safe node), a path in the evacuation graph, and a departure time. Each individual plan can be either directly derived from the optimization results or generated by introducing random perturbations. The MATSIM simulation engine uses the set of plans to simulate the movement of evacuees in the evacuation graph. It models each edge of the graph as a queue, which realistically simulates a real-world transportation network, in particular by considering congestion.

The first simulation experiment we conducted aims at studying the feasibility of the plan produced by the free flow model. Considering that FF does not produce a plan for each evacuee, we considered two scenarios, namely *Closest* and *Random Closest*. In the Closest scenario, each evacuee goes to the closest accessible safe node at its departure time, and we ensure that the total volume of evacuees leaving each area is the same as the one produced by the optimization. In the Random Closest scenario, we allow for more variation in the evacuees behaviors, and consider that 50% will go to the closest, 40% to one of the five closest, and

10% to a random safe node. In addition, we generate random departure times that depend on the earliest departure time of the neighboring areas and the latest departure time for the considered area.

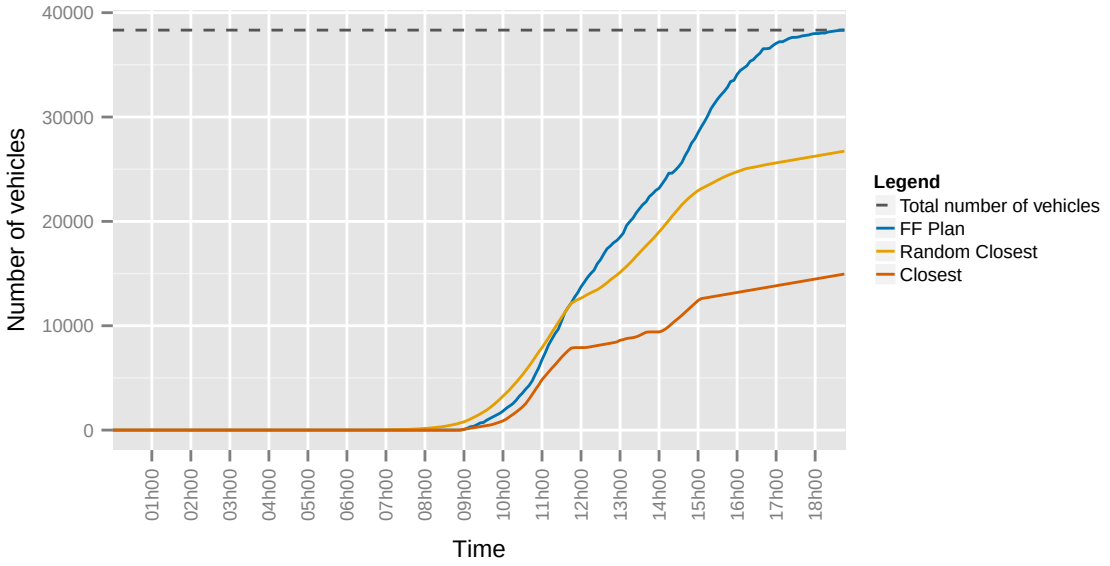



Figure 8: Comparison of evacuation profiles produced by the Free Flow (FF) model, Closest and Random Closest simulation scenarios on the HN instance.

Figure 8 presents the evacuation profile produced by the Free-Flow approach (FF Plan), and the Closest and Random Closest simulation scenarios. The dashed line represent the total number of vehicles to be evacuated in the considered area. This chart illustrates the fact that FF schedule is too optimistic when simulated. More generally, our experiments indicate that although FF predicts that all evacuees can reach a safe zone in the HN and HN-Ix instances, the simulation indicates that only 39% would be evacuated by the end of the planning horizon in the Closest scenario, and 66% in the Random Closest scenario. Detailed results area available in the Appendix.

The second experiment considers the evacuation plans produced by the path generation approach. Table 4 presents the number of evacuees reaching a safe node under different scenarios. The second column (Optimization Plan) corresponds to the results of the optimization, and the third (Simulation Plan) to the simulation of the plans produced by the optimization. The fourth column (Rush) corresponds to a scenario in which all evacuees from a same area leave at the first departure time produced by the optimization. The fifth column (Random Schedule) is a scenario in which the departure time of the evacuees is randomized depending on the earliest departure time of the neighboring areas and the latest departure time for the considered area. The sixth column (Random Plan) adds an additional level of randomization by considering that 50% of the evacuees will follow the plan, 40% will go to one on the five closest safe nodes, and 10% will go to a random safe node. Finally, the seventh column (Closest) represents a scenario in which evacuees depart as instructed but go to the closest safe node. The results illustrate that the evacuation plan produced by the CPG procedure is very close to the simulation and robust to evacuees behaviors. Of particular interest is the fact that Simulation Plan, Rush, Random Schedule, and Random Plan produce results within

3% of what was predicted by the Optimization Plan. In addition, our results indicate that CPG produces an evacuation schedule that allow to evacuate a majority of evacuees even in the Closest scenario.

| Instance | Opt. Plan | Sim. Plan | Rush | Rnd. Sched. | Rnd. Plan | Closest |
|---|---|---|---|---|---|---|
| **HN** | 100% | 100% | 100% | 98% | 97% | 78% |
| **HN-I1.1** | 100% | 100% | 100% | 98% | 95% | 78% |
| **HN-I1.2** | 100% | 100% | 100% | 97% | 97% | 75% |
| **HN-I1.4** | 100% | 100% | 100% | 98% | 95% | 77% |
| **HN-I1.7** | 97% | 97% | 98% | 93% | 96% | 79% |
| **HN-I2.0** | 92% | 92% | 93% | 90% | 93% | 76% |
| **HN-I2.5** | 90% | 90% | 91% | 87% | 90% | 69% |
| **HN-I3.0** | 87% | 87% | 87% | 85% | 89% | 63% |

Table 4: Percentage of evacuees reaching a safe node under different simulation scenarios.

Figure 9 illustrates the evacuation profiles produced by the CPG approach and the different simulation scenarios for the HN instance. These results illustrate the robustness of the plan produced by CPG: The curves representing the number of evacuees reaching safety are very close independently of the scenario considered. The only exception is Closest which generates more congestion and for which only 78% of the evacuees reach a safe node.

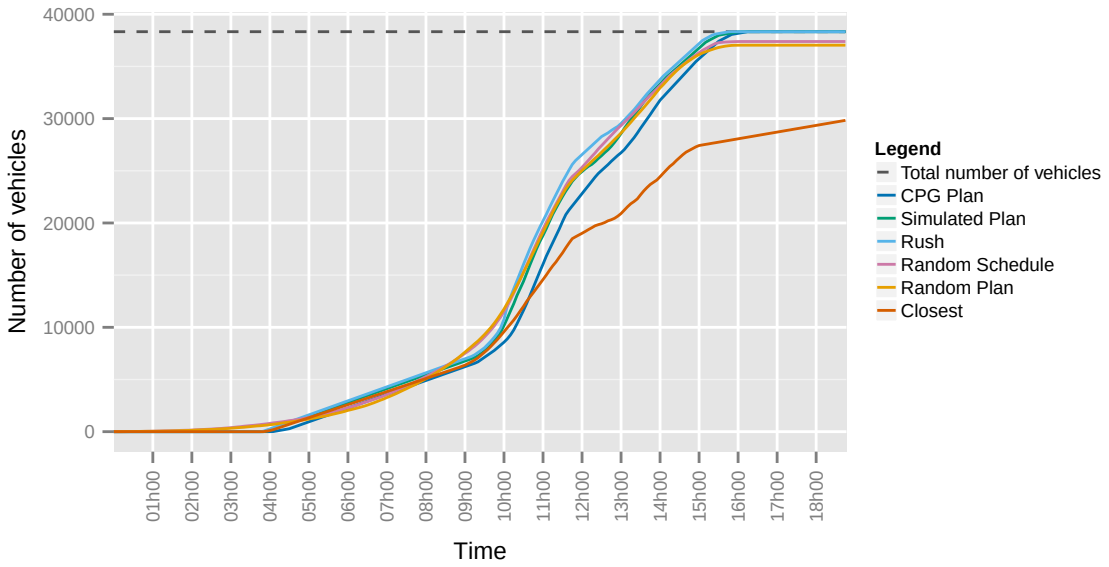



Figure 9: Comparison of evacuation profiles produced by the Path Generation (CPG) model, and different simulation scenarios on the HN instance.

# 7 Conclusions

This paper considered evacuation planning and scheduling, a critical aspect of disaster management and national security applications. It proposed a conflict-based path-generation approach whose key idea is to generate evacuation paths for evacuated areas iteratively and

optimize the evacuation over these paths in a master problem. Each new path is generated to remedy conflicts in the evacuation and adds new columns and a new row in the master problem. The algorithm was applied to massive flood scenarios in the Hawkesbury-Nepean floodplain (West Sydney, Australia) which require evacuating about 70,000 persons. Computational results show that the proposed path-generation approach is able to design evacuation plans for such large-scale scenarios in under 30 seconds, contrary to a traditional MIP approach which does not scale to this problem size. Of particular interest is the fact that the proposed approach reduces the number of variables from 4,500,000 in a MIP formulation to 30,000 in the case study.

To the best of our knowledge, this is the first scalable evacuation algorithm that conforms to evacuation methodologies and field requirements. Our evacuation algorithm can be used in strategic, tactical, and operational environments.

Our current work aims at improving the path generation using constraint programming to find new paths. Future work will also focus on microscopic modeling of the transportation system, the inclusion of loading curves for notification, and models of human behavior in evacuation settings.

**Acknowledgments** NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of1 Excellence program.

## Appendix: detailed simulation results for the free flow approach

Table 5 presents the percentage of evacuees reaching safety. The second column corresponds to the results of the optimization, and the third and fourth to the Closest and Random Closest simulation scenario.

| Instance | FF | Closest | Rnd. Closest |
|---|---|---|---|
| **HN** | 100% | 39% | 70% |
| **HN-I1.1** | 100% | 38% | 68% |
| **HN-I1.2** | 100% | 38% | 68% |
| **HN-I1.4** | 100% | 38% | 67% |
| **HN-I1.7** | 100% | 37% | 64% |
| **HN-I2.0** | 100% | 38% | 63% |
| **HN-I2.5** | 100% | 40% | 63% |
| **HN-I3.0** | 100% | 41% | 63% |

Table 5: Percentage of evacuees reaching a safe node when following the schedule produced by Free Flow (FF) under different simulation scenarios.

# References

[1] Alvelos, F. and Valrio De Carvalho, J. (2000). Solving multicommodity flow problems with branch-and-price. *Technical Report*.

[2] Barnhart, C., Hane, C., and Vance, P. (1997). Integer multicommodity flow problems. *Lecture Notes in Economics and Mathematical Systems*, 450:17–31.

[3] Barnhart, C., Hane, C. A., and Vance, P. H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326.

[4] Bish, D. R. and Sherali, H. D. (2013). Aggregate-level demand management in evacuation planning. *European Journal of Operational Research*, 224(1):79–92.

[5] Bretschneider, S. and Kimms, A. (2011). A basic mathematical model for evacuation problems in urban areas. *Transportation Research Part A: Policy and Practice*, 45(6):523–539.

[6] Bretschneider, S. and Kimms, A. (2012). Pattern-based evacuation planning for urban areas. *European Journal of Operational Research*, 216(1):57–69.

[7] Coffrin, C., Van Hentenryck, P., and Bent, R. (2011). Strategic stockpiling of power system supplies for disaster recovery. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE.

[8] Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287.

[9] Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors (2005). *Column Generation*. Mathematics of Decision Making. Springer.

[10] Hamacher, H. W. and Tjandra, S. A. (2001). Mathematical modelling of evacuation problems: A state of art. Technical report, Fraunhofer Institut für Techno und Wirtschaftsmathematik.

[11] Holmberg, K. and Yuan, D. (2003). A multicommodity network-flow problem with side constraints on paths solved by column generation. *INFORMS Journal on Computing*, 15(1):42–57.

[12] Lim, G. J., Zangeneh, S., Baharnemati, M. R., and Assavapokee, T. (2012). A capacitated network flow optimization approach for short notice evacuation planning. *European Journal of Operational Research*, 223(1):234–245.

[13] Lin, P., Lo, S., Huang, H., and Yuen, K. (2008). On the use of multi-stage time-varying quickest time approach for optimization of evacuation planning. *Fire Safety Journal*, 43(4):282–290.

[14] Liu, H. X., He, X., and Ban, X. (2007). A cell-based many-to-one dynamic system optimal model and its heuristic solution method for emergency evacuation. In *Proc. 86th Annual Meeting Transportation Res. Board*.

[15] Lu, Q., George, B., and Shekhar, S. (2005). Capacity constrained routing algorithms for evacuation planning: A summary of results. In Bauzer Medeiros, C., Egenhofer, M., and Bertino, E., editors, *Advances in Spatial and Temporal Databases*, volume 3633 of *Lecture Notes in Computer Science*, pages 291–307. Springer Berlin Heidelberg.

[16] Lu, Q., Huang, Y., and Shekhar, S. (2003). Evacuation planning: A capacity constrained routing approach. In Chen, H., Miranda, R., Zeng, D., Demchak, C., Schroeder, J., and Madhusudan, T., editors, *Intelligence and Security Informatics*, volume 2665 of *Lecture Notes in Computer Science*, pages 111–125. Springer Berlin Heidelberg.

[17] Lübbecke, M. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

[18] Massen, F., Deville, Y., and Hentenryck, P. V. (2012). Pheromone-based heuristic column generation for vehicle routing problems with black box feasibility. In *CPAIOR*, pages 260–274.

[19] MATSIM (2013). Matsim reference manual. `http://www.matsim.org/docs`.

[20] Richter, K.-F., Shi, M., Gan, H.-S., and Winter, S. (2013). Decentralized evacuation management. *Transportation Research Part C: Emerging Technologies*, 31(0):1–17.

[21] SES-NSW (2005). Hawkesbury nepean flood emergency sub plan. Technical report, State Emergency Service - New South Wales.