Received: 5 March 2020 | Revised: 21 July 2020 | Accepted: 23 July 2020 | Published on: 7 September 2020

DOI: 10.1002/net.21981

SPECIAL ISSUE ARTICLE

WILEY

Large-scale zone-based evacuation planning—Part I: Models and algorithms

Mohd. Hafiz Hasan[1] | **Pascal Van Hentenryck**[2]

[1]University of Michigan, Ann Arbor, Michigan,
[2]Georgia Institute of Technology, Atlanta, Georgia,

Correspondence
Pascal Van Hentenryck, Georgia Institute of Technology, Atlanta, GA 30332.
Email: pvh@isye.gatech.edu

Abstract

In zone-based evacuation planning, the region to evacuate is divided into zones, and each zone must be assigned a path to safety and departure times along the path. Zone-based evacuations are highly desirable in practice because they allow emergency services to communicate evacuation orders and to control the evacuation more precisely. Zone-based evacuations may also be combined with contraflows (to maximize the network capacities) and may impose additional constraints on the evacuation path (e.g., path convergence) and the departure times (e.g., non-preemption). This paper synthesizes existing models and algorithms for large-scale zone-based evacuation planning and complements them with some new ones to fill some of the gaps in the design space. Each model and algorithm is also extended to accommodate contraflows. A companion paper evaluates them on a real, large-scale case study, both from a macroscopic standpoint and through microscopic simulations under a variety of assumptions.

KEYWORDS

Benders decomposition, column generation, contraflow, evacuation planning and scheduling, mathematical optimization, non-preemptive and convergent evacuations, time-expanded graphs

1 | INTRODUCTION

Large-scale evacuations are often necessary and critical to the safety of residents in regions threatened by man-made or natural disasters like floods, hurricanes, and wildfires. According to a report by the International Federation of Red Cross and Red Crescent Societies [27], the first decade of the 21st century witnessed 7184 disasters around the world which affected a total of 2.55 billion people. These disasters accounted for the deaths of more than 1 million people and incurred $986 billion in economic losses. There is now an evacuation of 1000 or more people every 2 or 3 weeks in the United States alone.

Effective disaster management requires, among others, evacuation plans to ensure resources (e.g., the transportation network) are not completely overwhelmed by evacuees. Self-evacuations give each individual the freedom to choose their own evacuation routes, destinations, and times. However, they typically result in traffic congestion and associated delays that significantly increase the risk of casualties as individuals are being stranded in disaster affected areas. Therefore, it is crucial for emergency authorities to be equipped with centralized disaster management tools capable of generating and prescribing plans that guarantee an optimal utilization of evacuation resources. These plans aim at achieving specific goals such as maximizing the number of evacuees reaching safety or minimizing the overall evacuation time. Evacuation planning algorithms fulfill this need by producing prescriptive evacuation plans, that is, a set of operational instructions for authorities to manage and orchestrate large-scale evacuations. These instructions specify directions to be followed by evacuees, including routes from their homes to designated safe destinations and departure times, as well as the identification of roads to close in order to facilitate traffic flow. Prescriptive plans contrast with self-evacuations that are more difficult to control and may produce significant congestion.

Hamacher and Tjandra [11] distinguish between microscopic and macroscopic approaches to evacuation modeling. Microscopic approaches model individual characteristics of evacuees, their interactions with each other, and how these factors influence their movements. In contrast, macroscopic approaches aggregate evacuees and model their movements as flows in a network, making them much more amenable to optimization. Macroscopic models are often defined in terms of *flows over time* in order to capture capacity constraints more precisely. In particular, they typically use the concept of *time-expanded graphs* pioneered by Ford and Fulkerson [9].

This study is concerned with macroscopic approaches to prescriptive evacuation planning, although all results are validated using microscopic traffic simulations. Moreover, it focuses on *zone-based evacuations* in which all evacuees from the same residential zone are assigned a single evacuation route. Most emergency services rely on zone-based evacuations to facilitate the communication of evacuation plans, reduce confusion, increase compliance, and allow for a more reliable control of the evacuation. Indeed, zone-based evacuations are probably the only practical method for communicating instructions precisely to the population at risk. However, they are much more computationally challenging to plan and finding scalable algorithms has been one of the foci of recent research.

The core zone-based evacuation planning problem (ZEPP) considered in this study consists in assigning an evacuation path, as well as departure times, to each zone in the region. However, even when restricting attention to zone-based evacuations, several important design decisions remain to be taken. They include, but are not limited to, contraflows, convergent plans, and non-preemption.

- *Contraflows*, also known as lane reversals, are the idea of using inbound lanes for outbound traffic in evacuations. Several studies have suggested contraflow procedures as a viable method for increasing network capacity (e.g., [30, 32]).
- *Convergent plans* ensure that each evacuee coming to an intersection follows the same path subsequently. The rationale for convergent plans is that they eliminate forks from all evacuation paths and hence reduce driver hesitation at road intersections, which has been shown to be a significant source of delays [31]. Convergent paths also allow roads which are not part of the evacuation paths to be blocked, facilitating vehicular guidance and enforcement of the evacuation plans.
- *Non-preemptive* evacuations ensure that the evacuation of a zone, once it starts, proceeds without interruptions. Non-preemptive evacuations are also preferred by emergency services since they are easier to enforce.Each of these decisions has a significant impact, not only on the effectiveness of an evacuation (e.g., the number of evacuees reaching safety), but also on the computational properties of the optimization model and its potential solution techniques.

The purpose of this paper is to provide a systematic comparative study of zone-based evacuation planning, their design choices, their fidelity, and their computational performance. It synthesizes existing algorithms, complements them with some new ones to fill some of the gaps in the design space, and compares them on a real-life case study both at macroscopic and microscopic scales. The case study concerns the Hawkesbury-Nepean (HN) region located north-west of Sydney, Australia, and its associated time-expanded graph has 30 000 nodes and 75 000 arcs.

The benefits of this study are 3-fold:

1. It systematically evaluates, on a large case study and for the first time, a variety of zone-based evacuation planning algorithms both from macroscopic and microscopic viewpoints;
2. It quantifies, for the first time, the benefits and limitations of contraflows, convergent plans, and non-preemption, providing unique perspectives on how to deploy these algorithms in practice;
3. It highlights the approaches that are best suited to capture each of these design features and the computational burden they impose.

Finally, the paper addresses some avenues for future research that are not satisfactorily addressed in the literature yet are critical in practice.

The paper is organized in two parts. Part I reviews existing optimization algorithms and presents new ones. Part II evaluates them on a real, large-scale case study, from both a macroscopic standpoint and through microscopic simulations under a variety of assumptions. In Part I, Section 2 discusses some related work to give more context to the results presented in the paper. Section 3 outlines the key concepts in zone-based evacuation planning, including time-expanded graphs, contraflows, and convergent evacuations. Section 4 provides a mixed integer program (MIP) for non-convergent, preemptive evacuation planning with contraflows. This MIP introduces the main decision variables that appear in most approaches discussed in the paper. Section 5 presents a Benders decomposition approach for non-convergent, preemptive ZEPP, and Section 6 reviews the Benders decomposition approach for the convergent preemptive ZEPP originally proposed by Romanski and Van Hentenryck [28]. Section 7 presents the conflict-based path-generation heuristic algorithm proposed by Pillac et al. [26] for the non-convergent, preemptive ZEPP, whereas Section 8 reviews the column-generation approach to the non-convergent, non-preemptive ZEPP which was originally proposed by Pillac et al. [25] and improved upon by Hasan and Van Hentenryck [12]. Part II presents the case study and the experimental results from both a macroscopic and microscopic standpoint. It also gives some perspectives on directions for future research, knowledge gaps that need to be filled, and some related work to give more context to all the results in the paper.

2 | RELATED WORK

A comprehensive survey of evacuation planning is available in [2]. The goal of this section is to not duplicate this survey but to provide some context for this study. The importance of prescriptive evacuations, in which evacuation is orchestrated by a central authority as opposed to self-evacuations, is well recognized in the field, making it the focal point of various studies.

As mentioned in the introduction, Hamacher and Tjandra [11] distinguish between microscopic and macroscopic approaches to evacuation modeling. This paper presents macroscopic approaches that are evaluated using microscopic methods. *The majority of macroscopic approaches are flow-based. They solve the evacuation planning problem as a flow on a time-expanded graph.* For instance, Lu et al. [21,22] propose three heuristics to design an evacuation plan with multiple evacuation routes per evacuation node, minimizing the time of the last evacuation. They 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. [20] propose a heuristic algorithm for staged traffic evacuation (HASTE), whose main difference is the use of the cell transmission model (CTM) of Daganzo [7] to capture the flow of evacuees more precisely. Lim et al. [19] consider a short-notice regional evacuation maximizing the number of evacuees reaching safety weighted by the severity of the threat. They 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. Bretschneider and Kimms [4,5] focus on modeling the transportation network more precisely; they present a free-flow mathematical model over a detailed street network and include computational experiments on generated instances with a grid topology of up to 240 nodes, 330 edges, and a time horizon of 150 times steps.

Flow-based approaches are not zone-based and hence do not provide evacuation plans that comply with the procedures in place in most emergency services and local authorities. Assigning a routing and timing to every individual evacuee is unlikely to result in actionable plans. To the best of our knowledge, only a handful of studies design zone-based evacuation plans that produce both a set of evacuation routes and an evacuation schedule. Huibregtse et al. [16] propose a two-stage algorithm that first generates a set of evacuation routes and feasible evacuation times, and then assigns a route and time to each evacuated area using an ant colony optimization algorithm. The main difference with the present work is that the approach does not explicitly schedule the evacuation but relies on a third party simulator (EVAQ) to simulate the departure time of evacuees depending on the evacuation time decided for each area and to evaluate the quality of the solution. In later work, the authors studied the robustness of the produced solution [14], and strategies to improve the compliance of evacuees [15]. Bish and Sherali [3] present a model based on a CTM that assigns a single evacuation path to each evacuation node but fixes a response curve for each zone. Computational results include instances with up to 13 evacuation nodes, 2 safe nodes, and 72 edges. The conflict-based path generation method (CPG) and column generation (CG) methods presented in this paper were originally proposed by Pillac et al. [25,26]. Even et al. [8] take a different approach by introducing the convergent evacuation planning problem (CEPP) to remove forks from all evacuation routes and eliminate delays caused by diverging and crossing routes. They propose a two-stage approach to separate route design and evacuation scheduling. Their work served as the foundation of the algorithm by Romanski and Van Hentenryck [28], one of the algorithms described in detail in this paper (method BC). Andreas and Smith [1] consider a stochastic evacuation planning with convergent paths. Although not directly related, it is interesting to mention the work of Chen and Miller-Hooks [6] who provide an exact technique based on Benders decomposition to solve the Building Evacuation Problem with Shared Information. The problem is formulated as a MIP with shared information constraints to ensure evacuees departing from the same location at the same departure time receive common instructions. The MIP seeks to route evacuees from multiple locations in a building to exits such that total evacuation time is minimized.

3 | PROBLEM FORMULATION

Figure 1 shows an example of the evacuation scenario that is addressed by all methods presented in this work. Square 0 represents an evacuation node (e.g., a residential zone), triangles A and B represent safe nodes (final evacuation destinations), circles 1-3 represent transit nodes (road intersections), and arcs represent roads connecting the nodes. Times on each arc indicate when each road will become unavailable (e.g., due to being flooded), and the time on the evacuation node indicates the final deadline by which it must be evacuated. In this example, the evacuation deadline for node 0 is 13:00 since its last outgoing arc will be blocked at that time.

The evacuation scenario can be abstracted by a static evacuation graph $\mathcal{G} = (\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ where $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{S}$ are respectively the set of evacuation, transit, and safe nodes, and $\mathcal{A}$ is the set of all arcs. Each evacuation node $k \in \mathcal{E}$ is associated with its demand d_k representing the number of vehicles to be evacuated and its evacuation deadline f_k, whereas each arc $e \in \mathcal{A}$ is associated with its travel time s_e, its capacity u_e in terms of vehicles per unit time, and its block time f_e at which the road becomes unavailable. Figure 2 shows the static evacuation graph for the scenario of Figure 1. The evacuation node is labeled

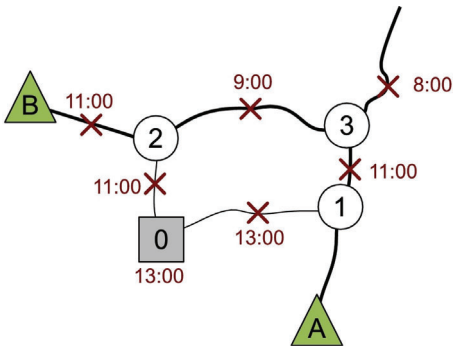

FIGURE 1 A sample evacuation scenario [Color figure can be viewed at wileyonlinelibrary.com]

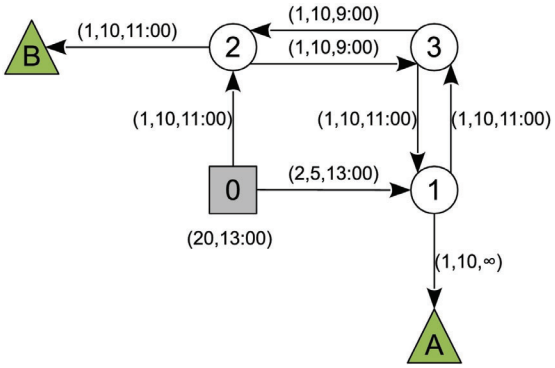

FIGURE 2 The static evacuation graph for the scenario in Figure 1 [Color figure can be viewed at wileyonlinelibrary.com]

with its demand and evacuation deadline whereas each arc is labeled with its travel time, capacity, and block time. Also note that the evacuation node has no incoming arcs and the safe nodes have no outgoing arcs.

To reason about traffic flows over time, the static graph is converted into a time-expanded graph $\mathcal{G}^x = (\mathcal{N}^x = \mathcal{E}^x \cup \mathcal{T}^x \cup \mathcal{S}^x, \mathcal{A}^x)$. The conversion is performed by first discretizing the time horizon $\mathcal{H}$ into time steps of identical length $t \in \mathcal{H}$, creating a copy of all nodes at each time step, and replacing each arc $e = (i, j)$ with corresponding arcs $e_t = (i_t, j_{t+s_e})$ for each time step that e is available. Figure 3 shows the time-expanded graph constructed from the static graph of Figure 2, where each arc is labeled with its capacity. Infinite capacity arcs are introduced connecting the evacuation and safe nodes at each time step to allow evacuees to wait at those nodes. Nodes that cannot be reached from either the evacuation or safe nodes within the time horizon are removed from the graph (they are grayed out in Figure 3).

An evacuation plan can then be defined to contain the following two components: (a) a set of evacuation paths, each represented by a sequence of connected nodes in the static graph from an evacuation node to a safe node, specifying the route to be taken by residents of each evacuation node to reach safety, and (b) an evacuation schedule indicating the number of vehicles that need to depart from each evacuation node at each time step $t \in \mathcal{H}$. The ZEPP can now be defined as follows.

> **Definition 1.** Given an evacuation graph $\mathcal{G}$, the ZEPP consists of finding an evacuation path from each evacuation zone to a safe node that maximizes the flow of evacuees to safe nodes, while satisfying the problem constraints.

Note that, in Figure 3, only one path can leave from the evacuation node in a zone-based evacuation plan.

3.1 | Contraflows

Contraflows are an important tool in evacuation planning and scheduling. To capture their benefits, this study assumes the existence of a subset $\mathcal{A}_c \subseteq \mathcal{A}$ of arcs in the static graph that may be used in contraflows during evacuations. The unique arc that goes in the opposite direction of arc $e \in \mathcal{A}_c$ is denoted by $\overline{e}$. The set $\mathcal{A}_c$ can then be partitioned into $\widehat{\mathcal{A}}_c$ and $\breve{\mathcal{A}}_c$ such that $\forall e \in \widehat{\mathcal{A}}_c : \overline{e} \in \breve{\mathcal{A}}_c$. Finally, $e_0 \in \mathcal{A}$ denotes the static edge associated with edge $e \in \mathcal{A}^x$ and $\delta^-(i)$ and $\delta^+(i)$ denote the set of incoming and outgoing arcs of node i, respectively.

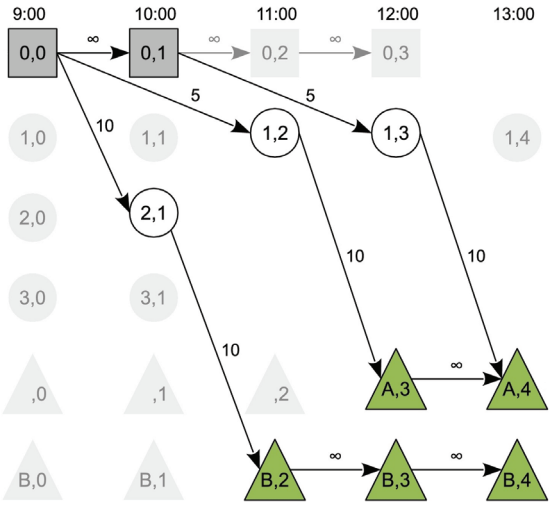

FIGURE 3 The time-expanded graph of the static graph in Figure 2 [Color figure can be viewed at wileyonlinelibrary.com]

3.2 | Convergent evacuations

Convergent paths reduce confusion and hesitation during an evacuation. They can be formally defined by the following definitions.

Definition 2. A graph $\mathcal{G} = (\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ is connected if, for all $k \in \mathcal{E}$, there exists a path from k to a safe node.

Definition 3. A graph $\mathcal{G} = (\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ is convergent if, for all $i \in \mathcal{E} \cup \mathcal{T}$, the outdegree of i is 1.

As stated by Even et al. [8], any connected evacuation graph $\mathcal{G}$ contains a connected and convergent subgraph $\mathcal{G}'$. If an evacuation graph is connected and convergent, each evacuation node has a unique path to a safe node. The convergent zone-based evacuation planning problem (C-ZEPP) is defined as follows.

Definition 4. Given a connected evacuation graph $\mathcal{G}$, the C-ZEPP consists of finding a convergent subgraph $\mathcal{G}'$ of $\mathcal{G}$ and a set of evacuee departure times that maximize the flow from evacuation nodes to safe nodes, while satisfying the problem constraints.

3.3 | Non-preemption and response curves

Non-preemptive evacuation plans are typically organized around the concept of response curves [24]. A response curve f is a function that models the number of evacuees departing from an evacuation node over time after an evacuation start time $t_0 \in \mathcal{H}$. The number $D_k(t)$ of evacuees departing from evacuation node k at time t is defined using a selected response curve f as follows:

$$D_k(t) = \begin{cases} 0 & \text{if } t < t_0 \\ f(t - t_0) & \text{if } t \geq t_0. \end{cases} \tag{1}$$

$D_k(t)$ can be used to specify a non-preemptive evacuation schedule for evacuation node k. Figure 4 shows $D_k(t)$ utilizing four different types of response curves. The step response curve, where evacuees depart at a constant rate after t_0 until a region is completely evacuated, is the response curve considered in this paper. In the figure, $t_0 = 120$ minutes.

Definition 5. Given an evacuation graph $\mathcal{G}$, the non-preemptive zone-based evacuation planning problem (NP-ZEPP) consists of finding an evacuation path from each evacuation zone to a safe node, a departure time, and a response curve that maximize the flow of evacuees to safe nodes and satisfy the problem constraints.

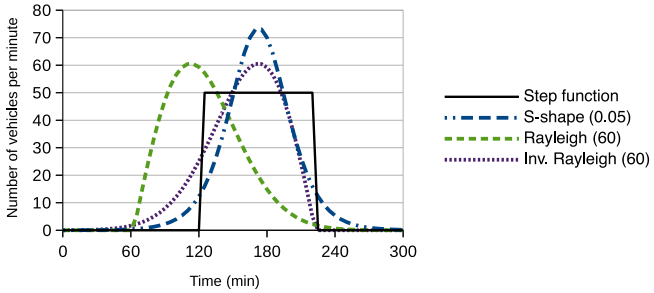

FIGURE 4 The number of evacuees departing an evacuation node k as a function of time, $D_k(t)$, using four different types of response curves *Source:* From Pillac et al. [25] [Color figure can be viewed at wileyonlinelibrary.com]

$$\max \quad \sum_{k\in\mathcal{E}^x} \sum_{e_t\in\delta^+(k)} \varphi_{e,k} \tag{2}$$

$$\text{s.t.} \quad \sum_{e\in\delta^+(k)} x_{e,k} = 1 \qquad \forall k \in \mathcal{E} \tag{3}$$

$$\sum_{e\in\delta^-(i)} x_{e,k} - \sum_{e\in\delta^+(i)} x_{e,k} = 0 \qquad \forall k \in \mathcal{E}, i \in \mathcal{T} \tag{4}$$

$$\sum_{e\in\delta^-(i)} \varphi_{e,k} - \sum_{e\in\delta^+(i)} \varphi_{e,k} = 0 \qquad \forall k \in \mathcal{E}, i \in \mathcal{T}^x \tag{5}$$

$$\sum_{k\in\mathcal{E}} \varphi_{e,k} \le u_e \qquad \forall e \in \mathcal{A}^x \setminus \mathcal{A}_c \tag{6}$$

$$\sum_{k\in\mathcal{E}} \varphi_{e,k} \le y_{e_0} u_e + (1 - y_{\bar{e}_0}) u_{\bar{e}} \qquad \forall e \in \mathcal{A}_c \tag{7}$$

$$\varphi_{e,k} \le u_e \cdot x_{e,k} \qquad \forall e \in \mathcal{A}^x, k \in \mathcal{E} \tag{8}$$

$$y_e + y_{\bar{e}} \ge 1 \qquad \forall e \in \hat{\mathcal{A}}_c \tag{9}$$

$$y_e \in \{0,1\} \qquad \forall e \in \mathcal{A}_c \tag{10}$$

$$\varphi_{e,k} \ge 0, x_{e,k} \in \{0,1\} \qquad \forall e \in \mathcal{A}^x, k \in \mathcal{E} \tag{11}$$

FIGURE 5 The MIP model for the ZEPP

The paper considers the ZEPP, C-ZEPP, and NP-ZEPP problems and algorithms to solve them. Each of these problems is considered with and without contraflows.

4 | THE BASIC MIP MODEL FOR THE ZEPP

This section presents a MIP model for the ZEPP. Model (2)-(11) in Figure 5 provides the intuition that serves as the basis for the more complex models presented subsequently. The decision variables of the model are as follows. The binary variable $x_{e,k}$ is equal to 1 if and only if edge $e \in \mathcal{A}$ belongs to the evacuation path for evacuation node k, and $\varphi_{e,k}$ is a continuous variable equal to the flow of evacuees from evacuation node k on edge $e \in \mathcal{A}^x$. To indicate which road should be used in contraflows, the binary variable y_e represents whether arc e is used in its normal direction ($y_e = 1$) or in contraflow ($y_e = 0$). Each road segment $(e, \bar{e})$ with $e \in \hat{\mathcal{A}}_c, \bar{e} \in \breve{\mathcal{A}}_c$ can then be utilized in one of three possible configurations: (a) $(y_e, y_{\bar{e}}) = (1,1)$ where both arcs are used in their normal directions, (b) $(y_e, y_{\bar{e}}) = (1,0)$ where arc $\bar{e}$ is used in contraflow, or (c) $(y_e, y_{\bar{e}}) = (0,1)$ where arc e is used in contraflow. The fourth case $(y_e, y_{\bar{e}}) = (0,0)$ will be excluded by one of the constraints.

Objective function (2) maximizes the number of evacuees reaching safety. Constraints (3) ensure that exactly one path is used to route the flow coming from evacuation nodes, while constraints (4) ensure the continuity of the path. Constraints (5) ensure flow conservation through the time-expanded graph. Constraints (6) enforce the capacity of each edge in the time-expanded graph. Constraints (7) enforce the capacity constraints on edges to allow for contraflows. They allocate to e the capacity of edge $\bar{e}$ whenever $\bar{e}$ is used in contraflow, and forbid any flow on e when it is used in contraflow. Constraints (8) ensure that there is no flow of evacuees coming from an evacuation node k if edge e is not part of the evacuation path for k. Constraints (9) prohibit the simultaneous use of e and $\bar{e}$ in contraflow.

Model (2)-(11) cannot be solved in reasonable time for the case study used in this paper which has approximately 30 000 nodes and 75 000 arcs. The computational difficulty comes from two interconnected components. The selection of the paths is a design component, whereas the scheduling of the evacuation is a multi-commodity flow problem. *The algorithms described in this paper address this computational challenge by separating these two aspects*. Observe also that the temporal aspects (i.e., when to schedule evacuees along a path) are an important and difficult aspect of the ZEPP. Finally, it is interesting to mention that additional requirements, such as convergent evacuations and non-preemption, may lead to elegant computational contributions that would not be possible otherwise.

5 | BENDERS DECOMPOSITION FOR THE ZEPP

This section presents an (approximate) Benders decomposition for the ZEPP. This Benders decomposition is referred to as the Benders non-convergent (BN) method in the rest of the paper. The restricted master problem (RMP) of the Benders decomposition selects evacuation paths that are then used in the subproblem (SP) for scheduling the flows of evacuees over time along these paths. It is important to note that the subproblem is a multi-commodity flow problem and hence its formulation is not totally unimodular. As a result, the Benders decomposition in the BN method solves a relaxation of the ZEPP where the integrality constraints on flow variables are relaxed. A final MIP is solved to obtain an integer solution to the subproblem.

As is traditional in Benders decomposition, the objective function values of the RMP and SP provide upper and lower bounds on the optimal solution to Model (2)-(11) without integrality constraints on the flow variables. When they converge, evacuation paths from the RMP and the evacuation schedule from the SP form an optimal solution. Otherwise, a Benders cut is generated from the solution of the SP and introduced into the RMP as an additional constraint to remove the current evacuation paths from the RMP's feasible region, after which the entire process is repeated.

5.1 | The restricted master problem

The RMP, depicted in Figure 6, finds evacuation paths for each evacuation zone. It operates on the static graph and its main decision variables are the binary variables $x_{e,k}$ of model (2)-(11). *In addition to the Benders cuts*, *the RMP also reasons about aggregate flows and aggregated capacities*, *an idea that was proposed by Romanski and Van Hentenryck* [28] *to obtain reasonable evacuation paths early in the series of algorithmic iterations*. In particular, variable $\psi_{e,k}$ represents the aggregate flow of evacuees from evacuation node k along arc e and arc capacities are aggregated over the time horizon in all of the RMP's capacity constraints. Finally, z_{RMP} is the RMP's objective function value and represents the number of evacuees reaching safety.

Constraint (13), together with objective function (12), maximizes the flow of evacuees from all evacuation nodes. Constraints (14) specify that exactly one path is generated for each evacuation node, and constraints (15) ensure that the one path requirement is preserved throughout the graph. Constraints (16) ensure that flow is conserved throughout the graph, and constraints (17) make sure that total flow from each evacuation node does not exceed its demand. Constraints (18) and (19) permit evacuee flow from evacuation node k on an arc only if the arc is selected for the evacuation path of k. Constraints (20) ensure that the flow from all evacuation nodes along an arc does not exceed the aggregate capacity for arcs that may not be used in contraflow, whereas constraints (21) do the same for arcs that may be used in contraflow. Finally, constraints (22) indicate that at most one arc in road segment $(e, \overline{e})$ with $e \in \widehat{\mathcal{A}}_c, \overline{e} \in \breve{\mathcal{A}}_c$ can be used in contraflow. To generate evacuation plans without contraflows, constraints (22) can be replaced with constraints $y_e + y_{\overline{e}} = 2 \;\; (e \in \mathcal{A}_c)$ to force all arcs to be used in their normal directions.

5.2 | The Benders subproblem

The SP, depicted in Figure 7, utilizes paths generated from the RMP together with the time-expanded graph $\mathcal{G}^x$ to generate an evacuation schedule that maximizes the number of evacuees reaching safety along those paths. The paths are specified by the values $\{\overline{x}_{e,k}\}$ and $\{\overline{y}_e\}$ for variables $\{x_{e,k}\}$ and $\{y_e\}$ in the RMP. The SP uses variable $\varphi_{e_t,k}$ to represent the flow of evacuees from evacuation node k along arc e_t in $\mathcal{G}^x$, and z_{SP} is the SP's objective function value.

Objective function (26) maximizes the flow of evacuees from all evacuation nodes. Constraints (27) enforce flow conservation throughout $\mathcal{G}^x$, whereas constraints (28) ensure that the total flow from each evacuation node does not exceed its demand. Constraints (29) and (30) permit flow only on the selected arcs for each evacuation node. Constraints (31) ensure that the total flow from all evacuation nodes along an arc does not exceed its capacity for arcs that may not be used in contraflow, whereas constraints (32) do the same for arcs that may be used in contraflow.

$$\max \quad z_{\text{RMP}} \tag{12}$$

$$\text{s.t.} \quad z_{\text{RMP}} \leq \sum_{k\in\mathcal{E}} \sum_{e\in\delta^+(k)} \psi_{e,k} \tag{13}$$

$$\sum_{e\in\delta^+(k)} x_{e,k} = 1 \qquad \forall k \in \mathcal{E} \tag{14}$$

$$\sum_{e\in\delta^+(i)} x_{e,k} \leq 1 \qquad \forall i \in \mathcal{T}, \forall k \in \mathcal{E} \tag{15}$$

$$\sum_{e\in\delta^-(i)} \psi_{e,k} - \sum_{e\in\delta^+(i)} \psi_{e,k} = 0 \qquad \forall i \in \mathcal{T}, \forall k \in \mathcal{E} \tag{16}$$

$$\sum_{e\in\delta^+(k)} \psi_{e,k} \leq d_k \qquad \forall k \in \mathcal{E} \tag{17}$$

$$\psi_{e,k} \leq x_{e,k} \sum_{t\in\mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A} \setminus \mathcal{A}_c, \forall k \in \mathcal{E} \tag{18}$$

$$\psi_{e,k} \leq x_{e,k} \sum_{t\in\mathcal{H}} (u_{e_t} + u_{\bar{e}_t}) \qquad \forall e \in \mathcal{A}_c, \forall k \in \mathcal{E} \tag{19}$$

$$\sum_{k\in\mathcal{E}} \psi_{e,k} \leq \sum_{t\in\mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A} \setminus \mathcal{A}_c \tag{20}$$

$$\sum_{k\in\mathcal{E}} \psi_{e,k} \leq y_e \sum_{t\in\mathcal{H}} u_{e_t} + (1 - y_{\bar{e}}) \sum_{t\in\mathcal{H}} u_{\bar{e}_t} \qquad \forall e \in \mathcal{A}_c \tag{21}$$

$$y_e + y_{\bar{e}} \geq 1 \qquad \forall e \in \mathcal{A}_c \tag{22}$$

$$\psi_{e,k} \geq 0 \qquad \forall e \in \mathcal{A}, \forall k \in \mathcal{E} \tag{23}$$

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

$$y_e \in \{0, 1\} \qquad \forall e \in \mathcal{A}_c \tag{25}$$

FIGURE 6 The restricted master problem for the BN method

$$\max \quad z_{\text{SP}} = \sum_{k\in\mathcal{E}} \sum_{e_t\in\delta^+(k)} \varphi_{e_t,k} \tag{26}$$

$$\text{s.t.} \quad \sum_{e_t\in\delta^-(i)} \varphi_{e_t,k} - \sum_{e_t\in\delta^+(i)} \varphi_{e_t,k} = 0 \qquad \forall i \in \mathcal{T}^x, \forall k \in \mathcal{E} \tag{27}$$

$$\sum_{t\in\mathcal{H}} \sum_{e_t\in\delta^+(k)} \varphi_{e_t,k} \leq d_k \qquad \forall k \in \mathcal{E} \tag{28}$$

$$\varphi_{e_t,k} \leq \bar{x}_{e,k} \cdot u_{e_t} \qquad \forall e \in \mathcal{A} \setminus \mathcal{A}_c, \forall t \in \mathcal{H}, \forall k \in \mathcal{E} \tag{29}$$

$$\varphi_{e_t,k} \leq \bar{x}_{e,k} \cdot (u_{e_t} + u_{\bar{e}_t}) \qquad \forall e \in \mathcal{A}_c, \forall t \in \mathcal{H}, \forall k \in \mathcal{E} \tag{30}$$

$$\sum_{k\in\mathcal{E}} \varphi_{e_t,k} \leq u_{e_t} \qquad \forall e \in \mathcal{A} \setminus \mathcal{A}_c, \forall t \in \mathcal{H} \tag{31}$$

$$\sum_{k\in\mathcal{E}} \varphi_{e_t,k} \leq \bar{y}_e \cdot u_{e_t} + (1 - \bar{y}_{\bar{e}}) \cdot u_{\bar{e}_t} \qquad \forall e \in \mathcal{A}_c, \forall t \in \mathcal{H} \tag{32}$$

$$\varphi_{e_t,k} \geq 0 \qquad \forall e_t \in \mathcal{A}^x, \forall k \in \mathcal{E} \tag{33}$$

FIGURE 7 The Benders subproblem for the BN method

5.3 | The Benders cuts

A Benders optimality cut is generated from the solution of the SP and added to the RMP as long as the objective function values of the RMP and SP do not converge. The cut is of the form

$$\begin{aligned} z_{\text{RMP}} \leq & \sum_{k\in\mathcal{E}} d_k \cdot \pi_k + \sum_{e\in\mathcal{A}\setminus\mathcal{A}_c} \sum_{t\in\mathcal{H}} u_{e_t} \sum_{k\in\mathcal{E}} x_{e,k} \cdot \pi_{e_t,k} \\ & + \sum_{e\in\mathcal{A}_c} \sum_{t\in\mathcal{H}} (u_{e_t} + u_{\bar{e}_t}) \sum_{k\in\mathcal{E}} x_{e,k} \cdot \pi_{e_{t,c},k} + \sum_{e\in\mathcal{A}\setminus\mathcal{A}_c} \sum_{t\in\mathcal{H}} u_{e_t} \cdot \pi_{e_t} \\ & + \sum_{e\in\mathcal{A}_c} \sum_{t\in\mathcal{H}} [y_e \cdot u_{e_t} + (1 - y_{\bar{e}}) \cdot u_{\bar{e}_t}] \cdot \pi_{e_{t,c}} \end{aligned} \tag{34}$$

where $\{\pi_k\}$, $\{\pi_{e_t,k}\}$, $\{\pi_{e_{t,c},k}\}$, $\{\pi_{e_t}\}$, and $\{\pi_{e_{t,c}}\}$ are the dual variables of constraints (28), (29), (30), (31), and (32), respectively. Since the SP is always feasible, Benders feasibility cuts are never generated.

5.4 | The Benders non-convergent algorithm

Algorithm 1 summarizes the entire BN algorithm which uses RMP($\mathcal{G}, \mathcal{H}$) to denote an optimal solution obtained from solving the RMP given static graph $\mathcal{G}$ and time horizon $\mathcal{H}$ as inputs, SP($\Psi, \mathcal{H}$) to denote an optimal solution of the SP given a solution to the RMP, Ψ, and time horizon $\mathcal{H}$ as inputs, and $z(\sigma)$ to denote the objective function value of a solution σ.

Algorithm 1. Benders non-convergent

$t^* \leftarrow \min\{t \in \mathcal{H} \mid z(\text{RMP}(\mathcal{G}, [0..t])) = z(\text{RMP}(\mathcal{G}, \mathcal{H}))\}$
$z_{\text{RMP}} \leftarrow z(\text{RMP}(\mathcal{G}, [0..t^*]))$
$z_{\text{SP}} \leftarrow z(\text{SP}(\text{RMP}(\mathcal{G}, [0..t^*]), \mathcal{H}))$
$z_{\text{SP, max}} \leftarrow z_{\text{SP}}$
while $z_{\text{RMP}} - z_{\text{SP, max}} \ldots \varepsilon$ **do.**
    Generate Benders cut from solution of SP and add it to RMP
    $z_{\text{RMP}} \leftarrow z(\text{RMP}(\mathcal{G}, \mathcal{H}))$
    $z_{\text{SP}} \leftarrow z(\text{SP}(\text{RMP}(\mathcal{G}, \mathcal{H}), \mathcal{H}))$
    $z_{\text{SP, max}} \leftarrow \max\{z_{\text{SP, max}}, z_{\text{SP}}\}$
Solve SP(RMP($\mathcal{G}, \mathcal{H}$), $\mathcal{H}$) with $\varphi_{e_t,k}$ integer $\forall e_t \in \mathcal{A}^x, \forall k \in \mathcal{E}$
return Evacuation paths from solution of RMP and evacuation schedule from solution of SP.

The BN algorithm begins with a procedure that searches for the tightest time horizon t^* preserving the optimal solution to the RMP, $z(\text{RMP}(\mathcal{G}, \mathcal{H}))$. This step was originally proposed by Even et al. [8] who found that a tighter time horizon produces better evacuation paths for the flow scheduling problem of their two-stage approach. The BN method adopts a similar strategy to seed the Benders decomposition. The procedure is implemented using a simple sequential search that solves RMP($\mathcal{G}, \mathcal{H}$) with progressively smaller values of $\mathcal{H}$ in search of t^*.

After this step, the algorithm proceeds to first solve the RMP to generate evacuation paths, and then the SP using the generated paths as input to generate an evacuation schedule. The minimum objective function value z_{RMP} of the RMP is then compared to the maximum objective function value $z_{\text{SP, max}}$ of the SP. If they do not converge (if their difference is larger than a convergence criterion ε set to 0), a Benders cut is generated using the dual variables from the SP solution and added to the RMP to remove the current evacuation paths from its feasible region. The process of solving the RMP and SP is then repeated until convergence. Since the Benders decomposition relaxes the integrality constraints on the flow variables, the subproblem is solved one more time as a MIP after convergence to obtain an integer solution to the subproblem.

6 | BENDERS DECOMPOSITION FOR CONVERGENT EVACUATION PLANNING

This section present the Benders decomposition of Romanski and Van Hentenryck [28] for the C-ZEPP, that is, for the convergent preemptive ZEPP. This Benders decomposition is referred to as the Benders convergent (BC) method in this paper. The BC method shares a lot of similarities with the BN method. However, the BC method imposes that the evacuation paths form a convergent graph and hence that the outdegree at each transit node is at most 1. *This convergence property has some fundamental consequences*: (1) *the BC method is exact since the subproblem becomes totally submodular*; (2) *it trivially supports contraflows*; *and* (3) *its computational performance is strong compared to all the other algorithms*.

6.1 | The restricted master problem

The RMP for the BC method is presented in Figure 8. *Because the paths are convergent*, *the model is considerably simpler*. *There is no need to track the origin of the flow* (*i.e.*, *the evaluation zone*) *and have different flow conservation constraints for each evacuation zone*. The model still uses a binary variable x_e to indicate whether an arc e is to be part of an evacuation path. But it uses a single variable ψ_e to represent the aggregate flow of evacuees along arc e over the time horizon. Arc capacities are again aggregated over the time horizon in the capacity constraints. Constraint (36) combined with objective function (35)

$$\max \quad z_{\text{RMP}} \tag{35}$$

$$\text{s.t.} \quad z_{\text{RMP}} \leq \sum_{k \in \mathcal{E}} \sum_{e \in \delta^+(k)} \psi_e \tag{36}$$

$$\sum_{e \in \delta^-(i)} \psi_e - \sum_{e \in \delta^+(i)} \psi_e = 0 \qquad \forall i \in \mathcal{T} \tag{37}$$

$$\sum_{e \in \delta^+(i)} x_e \leq 1 \qquad \forall i \in \mathcal{E} \cup \mathcal{T} \tag{38}$$

$$\psi_e \leq x_e \sum_{t \in \mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A} \tag{39}$$

$$\sum_{e \in \delta^+(k)} \psi_e \leq d_k \qquad \forall k \in \mathcal{E} \tag{40}$$

$$\psi_e \geq 0 \qquad \forall e \in \mathcal{A} \tag{41}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathcal{A} \tag{42}$$

FIGURE 8 The restricted master problem for the BC method

$$\max \quad z_{\text{SP}} = \sum_{k \in \mathcal{E}} \sum_{e_t \in \delta^+(k)} \varphi_{e_t} \tag{43}$$

$$\text{s.t.} \quad \sum_{e_t \in \delta^-(i)} \varphi_{e_t} - \sum_{e_t \in \delta^+(i)} \varphi_{e_t} = 0 \qquad \forall i \in \mathcal{T}^x \tag{44}$$

$$\varphi_{e_t} \leq \bar{x}_e \cdot u_{e_t} \qquad \forall e \in \mathcal{A}, \forall t \in \mathcal{H} \tag{45}$$

$$\sum_{e_t \in \delta^+(k)} \varphi_{e_t} \leq d_k \qquad \forall k \in \mathcal{E} \tag{46}$$

$$\varphi_{e_t} \geq 0 \qquad \forall e_t \in \mathcal{A}^x \tag{47}$$

FIGURE 9 The Benders subproblem for the BC method

maximizes the flow of evacuees leaving all evacuation nodes. Constraints (37) enforce flow conservation, whereas constraints (38) enforce the convergence of arcs selected by the evacuation paths. Constraints (39) permit flows only on selected arcs and ensure aggregate flow along them do not exceed their aggregate capacity. Finally, constraints (40) ensure that the total flow from each evacuation node does not exceed its demand.

6.2 | The Benders subproblem

The Benders problem for method BC, depicted in Figure 9, is again simpler due to path convergence and uses a variable φ_{e_t} to represent the flow of evacuees along arc e_t in $\mathcal{G}^x$. Objective function (43) maximizes the flow of evacuees across all evacuation nodes. Constraints (44) enforce flow conservation, constraints (45) permit flow only on arcs selected for evacuation paths and ensure that the flow does not exceed the arc's capacity, and constraints (46) ensure that the total flow leaving each evacuation node does not exceed its demand.

6.3 | The Benders cuts

The Benders optimality cuts are of the form

$$z_{\text{RMP}} \leq \sum_{e \in \mathcal{A}} x_e \sum_{t \in \mathcal{H}} u_{e_t} \cdot \pi_{e_t} + \sum_{k \in \mathcal{E}} d_k \cdot \pi_k \tag{48}$$

and use dual variables $\{\pi_{e_t}\}$ and $\{\pi_k\}$ associated with constraints (45) and (46) of the SP, respectively. Again, Benders feasibility cuts are never generated in this algorithm because the SP is always feasible.

6.4 | Pareto-optimal cuts

The convergence of Benders decomposition can be accelerated through the utilization of Pareto-optimal cuts [28], that is, cuts not dominated by any other Benders cut. The Magnanti-Wong method [23] is utilized to generate these stronger cuts (see for

$$\max \quad \sum_{k\in\mathcal{E}} \sum_{e_t\in\delta^+(k)} \varphi_{e_t} + \xi \cdot z_{\mathrm{SP}} \tag{49}$$

$$\text{s.t.} \quad \sum_{e_t\in\delta^-(i)} \varphi_{e_t} - \sum_{e_t\in\delta^+(i)} \varphi_{e_t} = 0 \qquad \forall i \in \mathcal{T}^x \tag{50}$$

$$\varphi_{e_t} + x_e \cdot u_{e_t} \cdot \xi \le x_e^0 \cdot u_{e_t} \qquad \forall e \in \mathcal{A}, \forall t \in \mathcal{H} \tag{51}$$

$$\sum_{e_t\in\delta^+(k)} \varphi_{e_t} + d_k \cdot \xi \le d_k \qquad \forall k \in \mathcal{E} \tag{52}$$

$$\varphi_{e_t} \ge 0 \qquad \forall e_t \in \mathcal{A}^x \tag{53}$$

FIGURE 10 The subproblem for generating Pareto-optimal cuts in the BN method

instance [29] for subsequent work on generating Pareto-optimal cuts). The method requires a core point, that is, a point located within the relative interior of the convex hull of the feasibility domain of the RMP's first-stage variable $\{x_e\}$. For this formulation, the core point utilized is simply $x_e^0 = \frac{1}{|\delta^+(i)+1|}$ for each arc $e = (i, j)$. The dual of the Magnanti-Wong problem (DMWP), which utilizes this core point and the optimal objective function value of the SP, z_{SP}, is solved to generate a Pareto-optimal cut. The model, depicted in Figure 10, uses a variable ξ, that is, the dual variable associated with the constraint ensuring that the objection function value of the Magnanti-Wong problem is equal to z_{SP}.

To generate the Pareto-optimal cut, coefficients $\{\pi_{e_t}\}$ and $\{\pi_k\}$ in cut (48) are taken from the dual variables of constraints (51) and (52), respectively instead of those from constraints of the SP.

6.5 | Contraflow extension

The BC algorithm proposed by Romanski and Van Hentenryck [28] did not consider contraflows, but it can be easily extended to support this functionality. *In fact*, *convergent evacuations make contraflows very easy*, as their tree structure guarantees that, for any road segment $(e, \overline{e})$ with $e \in \widehat{\mathcal{A}}_c, \overline{e} \in \breve{\mathcal{A}}_c$, if $x_e = 1$, then $x_{\overline{e}} = 0$. In other words, if an arc $e \in \widehat{\mathcal{A}}_c$ is in an evacuation path, the corresponding unique arc in the opposite direction $\overline{e}$ is not. This constraint makes it possible to use arc $\overline{e}$ in contraflow if arc $e \in \widehat{\mathcal{A}}_c$ is being used in an evacuation plan, since arc $\overline{e}$ is guaranteed not to be part of any other evacuation path by the convergence constraint. As a consequence, the BC algorithm is extended to allow contraflow as follows. Before the algorithm is executed, the capacities of all arcs $e \in \mathcal{A}_c$ are replaced with new capacities $u_{e_t,\mathrm{new}} = u_{e_t} + u_{\overline{e}_t}$. To identify where to use contraflows, it suffices to identify arcs $e \in \mathcal{A}_c$ with flows $\varphi_{e_t} > u_{e_t}$, meaning that the extra capacity afforded by using arc $\overline{e}$ in contraflow is necessary to achieve optimality.

7 | THE CPG METHOD

This section summarizes the heuristic algorithm originally presented by Pillac et al. [26] to solve the ZEPP and called the CPG method. The CPG method originated from an attempt to design a column-generation algorithm for the ZEPP. However, each new path creates a collection of variables, that is, the path variables and the associated flow variables, and these variables are linked as in constraints (8) of the MIP model. Since the duals of these constraints are not readily available, it did not appear easy to derive a column-generation algorithm at the time. Hence, the CPG mimics the behavior of a column-generation algorithm, but its pricing subproblem is a heuristic. More precisely, the CPG breaks down the evacuation planning problem into a subproblem (SP) responsible for generating evacuation paths and a RMP responsible for selecting paths and scheduling the evacuation. The method maintains a subset of critical evacuation nodes $\mathcal{E}' \subseteq \mathcal{E}$, that is, evacuation nodes not fully evacuated, and it alternates execution of the SP and RMP until $\mathcal{E}'$ is empty.

7.1 | The restricted master problem

The RMP of the CPG method, shown in Figure 11, selects an evacuation path for each evacuation node and schedules the evacuees over them to maximize the number of evacuees reaching safety. The paths are selected from a set of evacuation paths Ω' generated by the SP. The RMP uses a binary variable x_p to indicate whether a path $p \in \Omega'$ is selected for the evacuation plan, a continuous variable φ_p^t to represent the number of evacuees departing along path p at departure time t, and a continuous variable $\overline{\varphi}_k$ to represent the number of evacuees that cannot be evacuated at evacuation node k. In addition to these variables, $\Omega_k \subset \Omega'$ is the subset of evacuation paths for evacuation node k, $\omega(e) \subseteq \Omega'$ is the subset of paths that contain arc e, $\mathcal{H}_p \subseteq \mathcal{H}$ is

$$\max \quad \sum_{p\in\Omega}\sum_{t\in\mathcal{H}_p} \varphi_p^t \tag{54}$$

$$\text{s.t.} \quad \sum_{p\in\Omega_k} x_p = 1 \qquad \forall k \in \mathcal{E} \tag{55}$$

$$\sum_{p\in\Omega_k}\sum_{t\in\mathcal{H}_p} \varphi_p^t + \bar{\varphi}_k = d_k \qquad \forall k \in \mathcal{E} \tag{56}$$

$$\sum_{\substack{p\in\omega(e)\\ t-\tau_p^e\in\mathcal{H}_p}} \varphi_p^{t-\tau_p^e} \le u_{e_t} \qquad \forall e \in \mathcal{A}\setminus\mathcal{A}_c, \forall t \in \mathcal{H} \tag{57}$$

$$\sum_{\substack{p\in\omega(e)\\ t-\tau_p^e\in\mathcal{H}_p}} \varphi_p^{t-\tau_p^e} \le y_e \cdot u_{e_t} + (1-y_{\bar{e}})\cdot u_{\bar{e}_t} \qquad \forall e \in \mathcal{A}_c, \forall t \in \mathcal{H} \tag{58}$$

$$y_e + y_{\bar{e}} \ge 1 \qquad \forall e \in \mathcal{A}_c \tag{59}$$

$$\sum_{t\in\mathcal{H}_p} \varphi_p^t \le |\mathcal{H}_p| \cdot x_p \cdot u_p \qquad \forall p \in \Omega' \tag{60}$$

$$\varphi_p^t \ge 0 \qquad \forall p \in \Omega', \forall t \in \mathcal{H}_p \tag{61}$$

$$\bar{\varphi}_k \ge 0 \qquad \forall k \in \mathcal{E} \tag{62}$$

$$y_e \in \{0,1\} \qquad \forall e \in \mathcal{A}_c \tag{63}$$

$$x_p \in \{0,1\} \qquad \forall p \in \Omega' \tag{64}$$

FIGURE 11 The restricted master problem for the CPG method

$$\min \quad \sum_{k\in\mathcal{E}'}\sum_{e\in\mathcal{A}} c_e \cdot y_{e,k} \tag{66}$$

$$\text{s.t.} \quad \sum_{e\in\delta^-(i)} y_{e,k} - \sum_{e\in\delta^+(i)} y_{e,k} = 0 \qquad \forall i \in \mathcal{T}, \forall k \in \mathcal{E}' \tag{67}$$

$$\sum_{e\in\delta^+(k)} y_{e,k} = 1 \qquad \forall k \in \mathcal{E}' \tag{68}$$

$$y_{e,k} \in \{0,1\} \qquad \forall e \in \mathcal{A}, \forall k \in \mathcal{E}' \tag{69}$$

FIGURE 12 The path generation subproblem for the CPG method

the subset of time steps over which path p is usable, τ_p^e is the number of time steps required to reach arc e when traversing path p, and u_p is the capacity of path p.

Objective function (54) maximizes the flow of evacuees over all paths. Constraints (55) allow only one path from being selected per evacuation node, whereas constraints (56) ensures that the sum of evacuees who reach or who do not reach safety is equal to the demand at each evacuation node. Constraints (57) and (58) enforce the capacity of arcs that may not and may be used in contraflow, respectively. Constraints (59) prohibit the simultaneous use of arcs e and $\bar{e}$ in contraflow for road segment $(e,\bar{e})$ with $e \in \widehat{\mathcal{A}}_c, \bar{e} \in \breve{\mathcal{A}}_c$. Finally, constraints (60) allow for flows only on selected paths. To generate an evacuation plan that does not permit contraflow, constraints (59) are replaced with constraints (65) to ensure all arcs are only used in their normal directions.

$$y_e + y_{\bar{e}} = 2 \qquad \forall e \in \mathcal{A}_c \tag{65}$$

Observe that constraints (60) feature both variables φ_p^t and x_p. These constraints must be generated every time a new path is available and they make it difficult to obtain a traditional pricing subproblem since their duals are not available.

7.2 | The path generation subproblem

The SP utilizes a CPG heuristic to generate evacuation paths that potentially improve the objective function value of the RMP. These paths are generated by solving the multiple-origins, multiple-destinations shortest path problem shown in Figure 12.

The problem formulation utilizes a binary variable $y_{e,k}$ to indicate whether arc e belongs to the path generated for evacuation node k. Objective function (66) minimizes the total cost of all paths. Arc cost c_e is defined as a linear combination of an arc's travel time s_e, the number of times arc e is utilized in the current set of paths Ω', and the utilization of arc e in the current

solution:

$$c_e = \alpha_t \frac{s_e \cdot r}{\max_{e \in \mathcal{A}} s_e} + \alpha_c \frac{\sum_{\substack{p \in \Omega' \\ e \in p}} 1}{\mid \Omega' \mid} + \alpha_u \frac{\sum_{\substack{p \in \Omega' \\ e \in p}} \sum_{t \in \mathcal{H}_p} \varphi_p^t}{\sum_{t \in \mathcal{H}} u_{e_t}} \tag{70}$$

In Equation (70), α_t, α_c, and α_u are positive weights which sum to 1, and r is a random noise factor that is initialized to 1 and subsequently modified to $r \in [1-\varepsilon, 1+\varepsilon]$ depending on the number of iterations in which the objective function value of the RMP did not improve. The value ε is set to 0.50 in this study. Constraints (67) enforce path continuity whereas constraints (68) ensure only one path is generated for each critical node.

7.3 | The CPG algorithm

This CPG algorithm is summarized in Algorithm 2. PathGenerationSP ($\mathcal{E}$', Ω', Λ) denotes a subroutine to solve the SP with a set of critical evacuation nodes $\mathcal{E}$', a set of evacuation paths Ω', and an evacuation schedule Λ obtained from the solution of the RMP as inputs. EvacuationSchedulingMP(Ω') denotes a subroutine to solve the RMP using a set of evacuation paths Ω' as input. FindCriticalEvacuationNodes(Λ), as its name suggests, is a subroutine that identifies evacuation nodes not fully evacuated using an evacuation schedule Λ as input.

Algorithm 2. Conflict-based path generation

1: $\Omega' \leftarrow$ PathGenerationSP($\mathcal{E}, \emptyset, \emptyset$)
2: $\Lambda \leftarrow$ EvacuationSchedulingMP(Ω')
3: $\mathcal{E}' \leftarrow$ FindCriticalEvacuationNodes(Λ)
4: **while** $\mathcal{E}' \neq \emptyset$ **do**
5: $\quad \Omega' \leftarrow \Omega' \cup$ PathGenerationSP($\mathcal{E}', \Omega', \Lambda$)
6: $\quad \Lambda \leftarrow$ EvacuationSchedulingMP(Ω')
7: $\quad \mathcal{E}' \leftarrow$ FindCriticalEvacuationNodes(Λ)
8: $\Lambda \leftarrow$ SolveEvacuationSchedulingMP(Ω') with φ_p^t integer $\forall p \in \Omega', \forall t \in \mathcal{H}_p$
9: **return** Selected evacuation paths from solution of RMP and evacuation schedule Λ.

The algorithm begins by first solving the SP to generate an evacuation path for each evacuation node. The RMP is then solved to schedule the flow of evacuees over these paths. Critical evacuation nodes are then identified and stored in $\mathcal{E}$' and, as long as this set is not empty, the process of solving the SP to generate additional evacuation paths and the RMP to produce an evacuation schedule that maximizes the flow of evacuees is repeated. The algorithm terminates when $\mathcal{E}$' is empty or when a maximum number of iterations is reached (maximum number of iterations is set to 10 in this study). Upon completion, the RMP is solved one last time as an IP, where variables $\{\varphi_p^t\}$ are set to be integers, to produce an evacuation schedule with integral flow values. In all but the instance with the largest population, the CPG method produces evacuation plans very quickly.

8 | CG FOR EVACUATION PLANNING

This section presents the CG algorithm introduced by Pillac et al. [25] to solve the NP-ZEPP, that is, the CG method generates non-preemptive, non-convergent zone-based evacuation paths. *Interestingly*, *forbidding preemption makes it possible to design an exact CG approach*, *avoiding the difficulties faced by the CPG method*. The key idea underlying the CG is to generate *time-response evacuation plans* of the form $p = \langle P, f, t_0 \rangle$ where.

1. P is an evacuation path for a given zone k;
2. t_0 is the starting time of the evacuation along path P;
3. $f \in \mathcal{F}$ is a response curve from a set $\mathcal{F}$ of predefined response curves.The CG method also features a multi-objective function to minimize the overall evacuation time in addition to maximizing the number of evacuees reaching safety.

8.1 | The restricted master problem

The RMP selects time-response evacuation plans from a subset of feasible plans Ω' to maximize the number of evacuees reaching safety and minimize the overall evacuation time. The formulation uses a number of constants associated with the plans. In particular, c_p denotes the cost for selecting plan p, $\Omega_k \subseteq \Omega'$ is the subset of plans for evacuation node k, $\omega(e) \subseteq \Omega'$ is the subset

of plans that utilize arc e, and a_{p,e_t} denotes the flow of evacuees along arc e at time t induced by plan p (as prescribed by the response curve and the departure time). The cost c_p of plan p is defined as a function that applies a linear penalty on the arrival time of evacuees at the safe node and heavily penalizes the number of evacuees that cannot reach safety. More precisely, c_p is defined as:

$$c_p = \sum_{e \in p} \sum_{t \in \mathcal{H}} c_{e_t} \cdot a_{p,e_t} + \overline{c} \cdot \overline{a}_p \tag{71}$$

where $\overline{a}_p$ denotes the number of evacuees not reaching safety when executing plan p, and c_{e_t} and $\overline{c}$ are defined as follows:

$$c_{e_t} = c_{(i,j)_t} = \begin{cases} \frac{t}{|\mathcal{H}|} & \text{if } j \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

$$\overline{c} = 100 \max_{e \in \mathcal{A}, t \in \mathcal{H}} \{c_{e_t}\} \cdot \max_{k \in \mathcal{E}} \{d_k\} \tag{73}$$

$$\min \quad \sum_{p \in \Omega'} x_p \cdot c_p \tag{74}$$

$$\text{s.t.} \quad \sum_{p \in \Omega_k} x_p = 1 \qquad \forall k \in \mathcal{E} \tag{75}$$

$$\sum_{p \in \omega(e)} a_{p,e_t} \cdot x_p \leq u_{e_t} \qquad \forall e \in \mathcal{A}, \forall t \in \mathcal{H} \tag{76}$$

$$x_p \geq 0 \qquad \forall p \in \Omega' \tag{77}$$

$$r_p = c_p - \mathbf{a}_p^\intercal \boldsymbol{\pi} < 0 \tag{78}$$

FIGURE 13 The restricted master problem for the CG method

The RMP uses a binary variable x_p to indicate whether plan $p \in \Omega'$ is selected and is shown in Figure 13. It is essentially a set-covering problem with constraints on the arc capacities. The RMP first maximizes the number of evacuees reaching safety and then minimizes the overall evacuation time. Constraints (75) ensure that only one plan is selected for each evacuation node k and constraints (76) enforce all arc capacities. The RMP minimizes the overall cost which essentially causes the objective function (74) to be multi-objective and lexicographic. The formulation is a linear relaxation of the original RMP; after completion of the column-generation phase, the RMP will be solved as a MIP.

8.2 | The reduced cost formulation

To find a time-response evacuation plan p that can improve the current RMP, its reduced cost r_p must be negative, that is, .,

$$r_p = c_p - \mathbf{a}_p^\intercal \boldsymbol{p} < 0 \tag{79}$$

where $\mathbf{a}_p$ is the column of constraint coefficients of x_p and $\boldsymbol{p}$ is the vector of dual values from the optimal solution of the RMP. Letting $\{\pi_k\}$, and $\{\pi_{e_t}\}$ denote the dual variables of constraints (75) and (76), respectively, and substituting Equation (71) into Equation (78), the reduced cost can be formulated as:

$$\begin{aligned} r_p &= \sum_{e \in p} \sum_{t \in \mathcal{H}} c_{e_t} \cdot a_{p,e_t} + \overline{c} \cdot \overline{a}_p - \pi_k - \sum_{e \in p} \sum_{t \in \mathcal{H}} \pi_{e_t} \cdot a_{p,e_t} \\ &= -\pi_k + \overline{c} \cdot \overline{a}_p + \sum_{e \in p} \sum_{t \in \mathcal{H}} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t} \end{aligned} \tag{80}$$

8.3 | The pricing subproblem

The PSP is responsible for identifying a new time-response evacuation plan that satisfies the negative reduced cost criteria. The formulation of the PSP exploits some key characteristics of the reduced cost. First, *the reduced cost contains terms that are specific to a single time-response evacuation plan p*. Since the time-response evacuation plans are independent of each other, the PSP can also be solved independently for each evacuation node $k \in \mathcal{E}$ and for each predefined response curve $f \in \mathcal{F}$, allowing multiple PSPs to be solved concurrently in parallel. Moreover, since π_k does not depend on the path, finding a plan p with negative reduced cost is equivalent to finding an evacuation path P and a start time t_0 that minimize the last two terms of

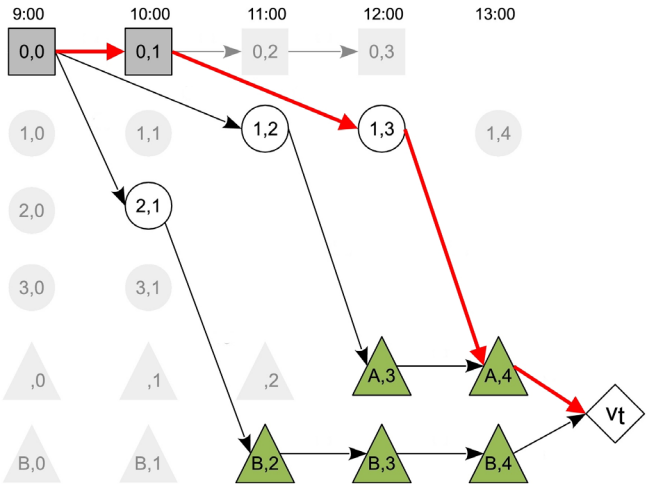

FIGURE 14 Path P^x in the extended time-expanded graph [Color figure can be viewed at wileyonlinelibrary.com]

Equation (80) for each $k \in \mathcal{E}$ and for each $f \in \mathcal{F}$. Denote the last two terms of Equation (80) as $\text{Cost}(P, t_0)$:

$$\text{Cost}(P, t_0) = \overline{c} \cdot \overline{a}_p + \sum_{e \in p} \sum_{t \in \mathcal{H}} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t}. \tag{81}$$

The second key observation is that the path P and evacuation start time t_0 that minimizes $\text{Cost}(P, t_0)$ can be obtained by applying a least-cost path algorithm on an extended time-expanded graph $\mathcal{G}^x$ with carefully defined arc costs. In particular, the formulation recognizes that a time-response evacuation plan follows the same path at each time step and hence the arc costs can be aggregated. The extension to the time-expanded graph involves the introduction of a virtual super sink, v_t, which all safe nodes $s \in \mathcal{S}^x$ are connected to with arcs $e_t \in \mathcal{A}^x_s = \{(s, v_t) \mid s \in \mathcal{S}^x\}$. Now denote by $\mathcal{A}^x_w$ the set of all infinite capacity arcs used to model evacuees waiting at the evacuation nodes. For a given evacuation node k and response curve f, a path P^x in $\mathcal{G}^x$ from evacuation node k_0 (evacuation node k and time 0) to v_t corresponds to a time-response evacuation plan $p = \langle P, f, t_0 \rangle$, where P is given by the sequence of nodes visited by P^x excluding v_t and t_0 is given by the time of the first non-waiting arc leaving $\mathcal{E}^x$. For instance, path P^x represented by the red colored arcs in Figure 14 corresponds to path $P = \langle 0, 1, \text{A} \rangle$ and evacuation start time $t_0 = 10{:}00$.

The cost of the combination (path, start time), $\text{Cost}(P, t_0)$, can be calculated by first assigning arc costs $c^{\text{sp}}_{e_t}$ to each arc $e_t \in \mathcal{A}^x$ as follows:

$$c^{\text{sp}}_{e_t} = \sum_{t'=t}^{|\mathcal{H}|} (c_{e_{t'}} - \pi_{e_{t'}}) \cdot f(t' - t) \qquad \forall e_t \in \mathcal{A}^x \backslash (\mathcal{A}^x_w \cup \mathcal{A}^x_s) \tag{82}$$

$$c^{\text{sp}}_{e_t} = \overline{c} \cdot (d_k - F(\mathcal{H}| - t)) \qquad \forall e_t \in \mathcal{A}^x_s \tag{83}$$

$$c^{\text{sp}}_{e_t} = 0 \qquad \forall e_t \in \mathcal{A}^x_w \tag{84}$$

Equation (82) aggregates future costs of arc e_t, should it be selected for a time-response evacuation plan, whereas Equation (83) accounts for the cost of evacuees not reaching safety for time-response evacuation plans which end with that arc.

With these arc cost definitions, $\text{Cost}(P, t_0)$ for a path P and an evacuation start time t_0 that corresponds to a path P^x can be calculated using Equation (85):

$$\text{Cost}(P, t_0) = \sum_{e_t \in P^x} c^{\text{sp}}_{e_t} \tag{85}$$

$$= \overline{c} \cdot (d_k - F(\mathcal{H}| - t)) + \sum_{e_t \in P^x \backslash \mathcal{A}^x_s} \sum_{t'=t}^{|\mathcal{H}|} (c_{e_{t'}} - \pi_{e_{t'}}) \cdot f(t' - t) \tag{86}$$

$$= \overline{c} \cdot \overline{a}_p + \sum_{e \in p} \sum_{t \in \mathcal{H}} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t} \tag{87}$$

Equations (86) and (87) show that the expansion of Equation (85) will eventually lead to the original equation for $\text{Cost}(P, t_0)$ in Equation (81).

With this formulation, the goal of the PSP, which is to find a path P and evacuation start time t_0 combination that minimizes $\text{Cost}(P, t_0)$, can be accomplished by finding a shortest path from k_0 to v_t in the extended time-expanded graph for each $k \in \mathcal{E}$ and $f \in \mathcal{F}$. As a result, an algorithm such as the Bellman-Ford algorithm to be applied to solve the PSP in polynomial time.

8.4 | The contraflow extension

The CG method can also be extended to produce an evacuation plan that allows for contraflows. It suffices to replace constraints (76) in the RMP with constraints (88) and (89), and to introduce additional constraints (90) and (91).

$$\sum_{p \in \omega(e)} a_{p,e_t} \cdot x_p \leq u_{e_t} \qquad \forall e \in \mathcal{A} \backslash \mathcal{A}_c, \forall t \in \mathcal{H} \tag{88}$$

$$\sum_{p \in \omega(e)} a_{p,e_t} \cdot x_p \leq y_e \cdot u_{e_t} + (1 - y_{\bar{e}}) \cdot u_{\bar{e}_t} \qquad \forall e \in \mathcal{A}_c, \forall t \in \mathcal{H} \tag{89}$$

$$y_e + y_{\bar{e}} \geq 1 \qquad \forall e \in \mathcal{A}_c \tag{90}$$

$$y_e \in [0, 1] \qquad \forall e \in \mathcal{A}_c \tag{91}$$

Constraints (88) enforce capacity on arcs that may not be used in contraflow, whereas constraints (89) do the same for arcs that may. Constraints (90) prevent arcs e and $\bar{e}$ from being used in contraflow simultaneously for road segment $(e, \bar{e})$ with $e \in \widehat{\mathcal{A}}_c, \bar{e} \in \breve{\mathcal{A}}_c$, and Constraints (91) apply a linear relaxation on variable y_e. Once the CG procedure has terminated, variable y_e is made binary, the final RMP is solved as a MIP, and the rest of the CG method remains unchanged.

8.5 | Elementary paths

The time-expanded graph $\mathcal{G}^x$ is by construction acyclic as its arcs only connect nodes at different time steps. As such, the shortest paths identified in the PSP are also acyclic. However, this fact does not preclude the PSP from generating paths that visit the same transit node in $\mathcal{G}^x$ at different time steps, as there are no restrictions enforced in the shortest path algorithms preventing such paths from being generated. Whereas such paths are acyclic in $\mathcal{G}^x$, their corresponding counterparts in the static graph $\mathcal{G}$ contain cycles, as they visit the same transit node more than once.

These cyclic paths are called non-elementary (they visit the same node multiple times), and an example of such a non-elementary path is shown in Figure 15. Non-elementary paths in the static graph are undesirable in real evacuations, as they give evacuees the impression that the evacuation plans are suboptimal and reduce trust in emergency services. However, when the CG algorithm is applied to the real case study, about 44% of the generated evacuation paths are not elementary.

This section outlines the pricing subproblem proposed by Hasan and Van Hentenryck [12] that only generates time-response evacuation plans with elementary paths. Let $\Lambda(i)$ denote the set of time-expanded nodes in $\mathcal{G}^x$ for a node $i \in \mathcal{T}$, that is, $\Lambda(i) = \{i_t \,|\, t \in \mathcal{H}\}$. A path P^x in $\mathcal{G}^x$ corresponds to an elementary path P in $\mathcal{G}$ if and only if P^x visits at most a single node in $\Lambda(i)$ for each node $i \in \mathcal{T}$. As a result, instead of finding a least-cost path, the revised PSP must find a least-cost path that is also an elementary path in the static graph. Figure 16 depicts the new formulation of the pricing problem. The formulation uses binary decision variable x_{e_t} to indicate whether edge e_t should be selected as part of the shortest path. Objective function (92) minimizes the total cost of the path. Constraint (93) specifies that exactly one path should emanate from source node k_0, whereas constraint (95) ensures the path ends at super sink node v_t. Constraints (94) enforce path continuity at every node other than the source and super sink. Finally, constraints (96) guarantee that each transit node is visited by the path at most once throughout the entire time horizon.

This version of the PSP is a shortest path problem with resource constraints [17] which is known to be NP-hard [10]. In this particular formulation, the resources are simply the unit "visited" resources associated with each transit node in $\mathcal{G}^x$. The set of all time steps of a particular transit node, $\{i_t \,|\, i \in \mathcal{T}, \forall t \in \mathcal{H}\}$, is allocated only one unit of this "visited" resource, and the resource is completely consumed if this node were to be visited by a path. For the case study in this paper, this constrained shortest-path problem must be solved repeatedly for a very large graph.

Whereas solving formulation (92)–(97) using a MIP solver will result in the desired shortest elementary path, the hybrid strategy proposed by Hasan and Van Hentenryck [12] is capable of obtaining these paths faster. The hybrid strategy combines the above formulation with a k-shortest-path-based algorithm based on the implementation of Jimenez and Marzal's Recursive Enumeration Algorithm (REA) [18]. This algorithm incrementally generates a kth-shortest path based on information of the $(k-1)$ shortest paths. It can be used to find the shortest elementary path by first generating the shortest path (setting $k = 1$). If

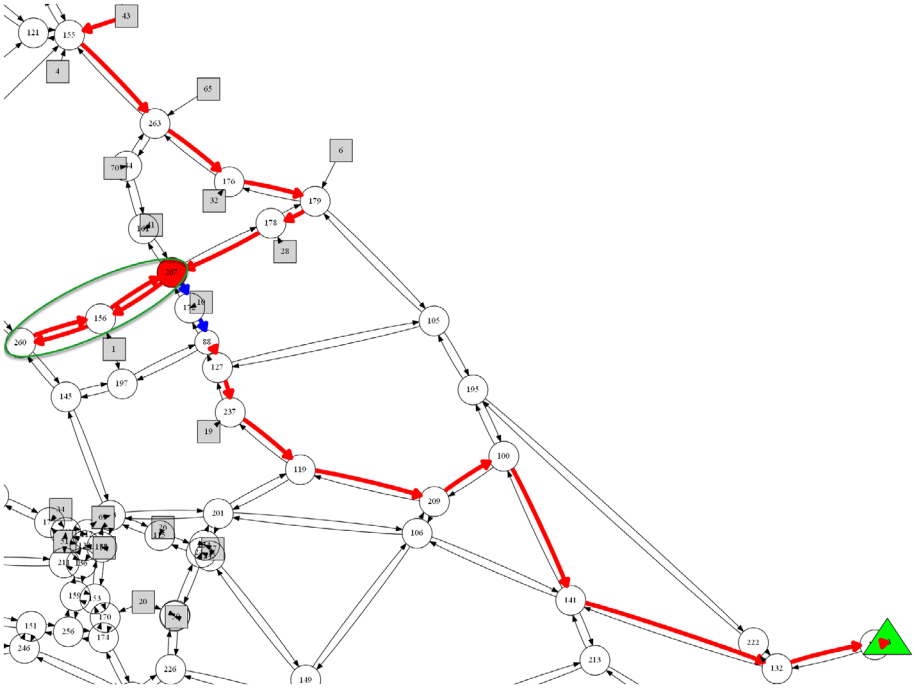

FIGURE 15 An example of non-elementary path [Color figure can be viewed at wileyonlinelibrary.com]

$$\min \quad \sum_{e_t \in \mathcal{A}^x} c^{\text{sp}}_{e_t} \cdot x_{e_t} \tag{92}$$

$$\text{s.t.} \quad \sum_{e_t \in \delta^+(k_0)} x_{e_t} = 1 \tag{93}$$

$$\sum_{e_t \in \delta^-(i)} x_{e_t} - \sum_{e_t \in \delta^+(i)} x_{e_t} = 0 \qquad \forall i \in \mathcal{N}^x \setminus \{k_0, v_t\} \tag{94}$$

$$\sum_{e_t \in \delta^-(v_t)} x_{e_t} = 1 \tag{95}$$

$$\sum_{i_t \in \Lambda(i)} \sum_{e_t \in \delta^+(i_t)} x_{e_t} \leq 1 \qquad \forall i \in \mathcal{T} \tag{96}$$

$$x_{e_t} \in \{0, 1\} \qquad \forall e_t \in \mathcal{A}^x \tag{97}$$

FIGURE 16 The pricing subproblem with elementary paths

the path is elementary, the algorithm terminates. Otherwise, the next shortest path is generated by the REA (by incrementing k by 1), and the elementary check is applied on this path. This process is repeated until an elementary path is obtained.

Computational experiments on the case study show that the k-shortest-path-based algorithm is extremely fast at finding shortest elementary paths when the required k values are relatively small ($k < 10^5$). Unfortunately, in some rare instances, the value of k required to obtain an elementary path is extremely large (in the millions), and under these circumstances, solving the MIP formulation produces faster results. Therefore, the hybrid strategy combines both methods by first utilizing the k-shortest-path-based algorithm to find shortest elementary paths up to a threshold value for k (in this study, the threshold is set to 10^5). If this k-threshold is reached and an elementary path is yet to be found, the MIP formulation is solved. This hybrid strategy exploits the strengths of both methods and is extremely effective at identifying shortest elementary paths quickly in almost all cases.

9 | CLEARANCE TIME MINIMIZATION

Of the four methods presented, only the CG method has a multi-objective function to minimize the total evacuation time in addition to maximizing the number of evacuees reaching safety. The BN, BC, and CPG methods only optimize for the latter goal. However, evacuation authorities are also deeply interested in the minimum clearance time, that is, the smallest amount of

time to evacuate an entire region. A precise definition of minimum clearance time, h^*, is as follows:

$$h^* = \min \left\{ t \in \mathcal{H} \mid z(\text{EPP}(\mathcal{G}, [0..t])) = \sum_{k \in \mathcal{E}} d_k \right\} \tag{98}$$

where $\text{EPP}(G, \mathcal{H})$ denotes the optimal solution obtained from an EPP formulation given static graph $\mathcal{G}$ and time horizon $\mathcal{H}$ as inputs. This section shows how to obtain the minimum clearance time for each method.

9.1 | Benders non-convergent and convergent methods

The BN and BC methods each consist of an RMP and an SP which generate upper and lower bounds on the objective function value. As proposed by Romanski and Van Hentenryck [28], a lower bound $h^\dagger$ on the minimum clearance time can first be obtained by performing a binary search over the time horizon using just the RMP. Next, a sequential search using the full BN or BC method can be used to find h^*, beginning from its lower bound $h^\dagger$. This approach seems preferable over a binary search for the second stage as h^* is very likely closer to the lower bound $h^\dagger$, and hence the second part of the algorithm will converge faster by starting a sequential search from that time. Algorithm 3 summarizes the entire approach.

Algorithm 3. Clearance time minimization for BN and BC methods

1: $h^\dagger \leftarrow \min \left\{ t \in \mathcal{H} \mid z(\text{RMP}(\mathcal{G}, [0..t])) = \sum_{k \in \mathcal{E}} d_k \right\}$
2: $h^* \leftarrow \min \left\{ t \in [h^\dagger..\mathcal{H}] \mid z(\text{EPP}(\mathcal{G}, [0..t])) = \sum_{k \in \mathcal{E}} d_k \right\}$
3: **return** h^*, evacuation paths from solution of RMP, and evacuation schedule from solution of SP.

9.2 | CPG method

Since the CPG method does not maintain upper and lower bounds on the objective function value of the EPP, the clearance time can be performed by a binary search over the time horizon of the full CPG method.

9.3 | CG method

Even though the CG method's multi-objective function minimizes total evacuation time, the quantity is not equivalent to clearance time. Clearance time is equivalent to the time at which the last evacuee arrives at its safe node, and this is not the quantity being minimized in the objective function. Minimizing total evacuation time might result in the minimal clearance time, but it might also produce suboptimal clearance times as the penalty incurred by the late arrival of the last evacuee could possibly be diluted by early arrival costs. Application of a binary search over the time horizon is an option, however this paper did not resort to this approach due to the significant run times of the CG method. Therefore, the minimum clearance time experiments only report the arrival time of the last evacuee produced by the CG method while fully recognizing that it might be suboptimal as the method's objective function does not explicitly minimize clearance times.

10 | CONCLUSION

This paper presented a systematic study of large-scale zone-based evacuation planning. It introduced the core ZEPP that consists in assigning an evacuation path, as well as departure times, to each zone in the region that needs to be evacuated. It also discussed critical additions to the core problem that include contraflows, convergent plans, and non-preemption. Part II of the paper evaluates them on a real, large-scale case study, both from a macroscopic standpoint and through microscopic simulations under a variety of assumptions, providing some unique perspectives on the strengths and weaknesses of each approach and the implications of evacuation functionalities.

ACKNOWLEDGMENTS

We would like to express our deep gratitude to Peter Cinque (New South-Wales State Emergency Services) and Peter Liehn (then at NICTA) for their leadership in the NICTA evacuation project, the foundation of this work. Manuel Cebrian, Caroline Even, Victor Pillac, and Andreas Schutt drove the development of many of the algorithms presented here and have been truly amazing collaborators. Finally, we would like to thank the reviewers for their detailed reading of the paper and their numerous suggestions, which improved the paper significantly.

ORCID

Pascal Van Hentenryck https://orcid.org/0000-0001-7085-9994

REFERENCES

[1] A. Andreas and J.C. Smith, *Decomposition algorithms for the design of a nonsimultaneous capacitated evacuation tree network*, Networks **53** (2009), 91–103.
[2] V. Bayram, *Optimization models for large scale network evacuation planning and management: A literature review*, Surv. Oper. Res. Manag. Sci. **21** (2016), 63–64.
[3] D. Bish and H. Sherali, *Aggregate-level demand management in evacuation planning*, Eur. J. Oper. Res. **224** (2013), 79–92.
[4] S. Bretschneider and A. Kimms, *A basic mathematical model for evacuation problems in urban areas*, Transp. Res. Part A: Pol. Pract. **45** (2011), 523–539.
[5] S. Bretschneider and A. Kimms, *Pattern-based evacuation planning for urban areas*, Eur. J. Oper. Res. **216** (2012), 57–69.
[6] L. Chen and E. Miller-Hooks, *The building evacuation problem with shared information*, Naval Res. Logist. **55** (2008), 363–376.
[7] C. Daganzo, *The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory*, Transp. Res. Part B: Methodol. **28** (1994), 269–287.
[8] C. Even, V. Pillac, and P. Van Hentenryck, "*Convergent plans for large-scale evacuations*," *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Cambridge, MA: AAAI Press, 2015, pp. 1121–1127.
[9] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
[10] M. Gary and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, CA, 1979.
[11] H. Hamacher and S. Tjandra, *Mathematical modelling of evacuation problems: A state of the art*, Pedestr. Evac. Dynam. **135**:(2002), 227–266.
[12] M.H. Hasan and P. Van Hentenryck, *A column-generation algorithm for evacuation planning with elementary paths*, Proceedings of the 23rd International Conference on Principles and Practice of Constraint Programming (CP-2017), Melbourne, Australia, August 2017.
[13] M.H. Hasan and P. Van Hentenryck, *Large-scale zone-based evacuation planning—Part II: Macroscopic and microscopic evaluations*, Networks (in press) (2020).
[14] O. Huibregtse, M. Bliemer, and S. Hoogendoorn, *Analysis of near-optimal evacuation instructions*, Procedia Eng. **3** (2010), 189–203.
[15] O. Huibregtse, A. Hegyi, and S. Hoogendoorn, *Blocking roads to increase the evacuation efficiency*, J. Adv. Transp. **46** (2012), 282–289.
[16] O. Huibregtse, S. Hoogendoorn, A. Hegyi, and M. Bliemer, *A method to optimize evacuation instructions*, OR Spectrum **33** (2011), 595–627.
[17] S. Irnich and G. Desaulniers, "*Shortest path problems with resource constraints*," *Column Generation*, G. Desaulniers, J. Desrosiers, and M.M. Solomon (eds), New York: Springer, 2005, pp. 33–65.
[18] V.M. Jiménez and A. Marzal, "*Computing the K shortest paths: A new algorithm and an experimental comparison*," *Algorithm Engineering*, J.S. Vitter and C.D. Zaroliagis(eds), Berlin, Heidelberg: Springer, 1999, pp. 15–29.
[19] G. Lim, S. Zangeneh, M.R. Baharnemati, and T. Assavapokee, *A capacitated network flow optimization approach for short notice evacuation planning*, Eur. J. Oper. Res. **223** (2012), 234–245.
[20] H.X. Liu, X. He, and X. Ban, "*A cell-based many-to-one dynamic system optimal model and its heuristic solution method for emergency evacuation*," *Proc. 86th Annual Meeting Transportation Res. Board*, Washington, D.C: Transportation Research Board, 2007, pp. 1–20.
[21] Q. Lu, B. George, and S. Shekhar, "*Capacity constrained routing algorithms for evacuation planning: A summary of results*," *Advances in Spatial and Temporal Databases*, C.B. Medeiros, M.J. Egenhofer, and E. Bertino (eds), Berlin, Heidelberg: Springer, 2005, pp. 291–307.
[22] Q. Lu, Y. Huang, and S. Shekhar, "*Evacuation planning: A capacity constrained routing approach*," *Intelligence and Security Informatics*, H. Chen, R. Miranda, D. Zeng, C. Demchak, J. Schroeder, and T. Madhusudan (eds), Berlin, Heidelberg: Springer, 2003, pp. 111–125.
[23] T. Magnanti and R. Wong, *Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria*, Oper. Res. **29** (1981), 464–484.
[24] A. Pel, M. Bliemer, and S. Hoogendoorn, *A review on travel behaviour modelling in dynamic traffic simulation models for evacuations*, Transportation **39** (2012), 97–123.
[25] V. Pillac, M. Cebrian, and P. Van Hentenryck, *A column-generation approach for joint mobilization and evacuation planning*, Constraints **20** (2015), 285–303.
[26] V. Pillac, P. Van Hentenryck, and C. Even, *A conflict-based path-generation heuristic for evacuation planning*, Transp. Res. Part B **83** (2016), 136–150.
[27] Red Cross, *World Disasters Report 2010—Focus on Urban Risk*, Int. Federation Red Cross Red Crescent Societies, Geneva, 2010.
[28] J. Romanski and P. Van Hentenryck, "*Benders decomposition for large-scale prescriptive evacuations*," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Cambridge, MA: AAAI Press, 2016, pp. 3894–3900.
[29] H.D. Sherali and B.J. Lunday, *On generating maximal nondominated Benders cuts*, Ann. Oper. Res. **210** (2013), 57–72.
[30] G. Theodoulou and B. Wolshon, *Alternative methods to increase the effectiveness of freeway contraflow evacuation*, Transp. Res. Rec.: J. Transp. Res. Board **1865** (2004), 48–56.
[31] F.F. Townsend, *The Federal Response to Hurricane Katrina: Lessons Learned*, White House, Washington, DC, 2006.
[32] B. Wolshon, *"One-way-out": Contraflow freeway operation for hurricane evacuation*, Nat. Hazards Rev. **2** (2001), 105–112.

How to cite this article: Hafiz Hasan M, Van Hentenryck P. Large-scale zone-based evacuation planning—Part I: Models and algorithms. *Networks.* 2021;77:127–145. https://doi.org/10.1002/net.21981