Proceedings of the 55th Hawaii International Conference on System Sciences | 2022

Large-scale Zone-based Evacuation Planning: Generating Convergent and Non-Preemptive Evacuation Plans via Column Generation

Jorge A. Huertas
Georgia Institute of Technology,
Atlanta, GA, USA
huertas.ja@gatech.edu

Pascal Van Hentenryck
Georgia Institute of Technology,
Atlanta, GA, USA
pascal.vanhentenryck@isye.gatech.edu

Abstract

In zone-based evacuations, the evacuated region is divided into zones, and vehicles follow the single evacuation path assigned to their corresponding zone. Ideally, these evacuation paths converge at intersections to reduce driver hesitation; and non-preemptive schedules ensure that the evacuation of a zone proceeds without interruptions once it starts. We present a column-generation algorithm that produces, for the first time, convergent and non-preemptive evacuation plans in real large-scale evacuation scenarios. Furthermore, we compare our algorithm against existing models that produce convergent paths or non-preemptive schedules separately. Finally, we use a traffic simulator to evaluate the quality of the generated plans in real-world settings.

1. Introduction

The traditional disaster management cycle consists of four primary phases: preparedness, mitigation, response, and recovery [1]. Upon the threat of a man-made or natural disaster, evacuation planning plays a critical role in the preparedness and response phases [2]. Therefore, when dealing with large-scale evacuations, emergency services require actionable evacuation plans that can be clearly communicated and controlled by authorities in these two phases.

Actionable evacuation plans require authorities to clearly communicate the evacuation paths and the evacuation schedule. To achieve this, a common practice is to use *zone-based evacuation plans* in which the evacuated region is divided into smaller zones, and a single evacuation path is assigned to each one of them [3]. Thus, all the vehicles from the same zone should follow the same assigned evacuation path, that conducts to a safe zone. Additionally, authorities often use *non-preemptive schedules*, which guarantee that the departure of vehicles from a zone is never interrupted once its evacuation starts [4]. This type of schedules are also easier to enforce and control.

Furthermore, to better control the development of the evacuation, erratic behavior from evacuees must be reduced. If two evacuation paths meet at the same intersection but lead to different places, driver hesitation will increase, resulting in additional congestion [5]. For this reason, authorities often prefer to use *convergent paths* that merge when they meet at intersections [6]. Convergent paths allow authorities to appropriately close access to unoccupied roads at strategic locations so that the flow of vehicles proceeds as planned.

The most challenging question when planning an evacuation is how to deal with the congestion. If poorly handled, arising congestion can delay evacuations for several hours, causing evacuees to remain at threat in the affected area [7]. One way to control its appearance is by conservatively setting departure rates in the non-preemptive schedule so that the network is never used above its capacity [8]. Another way is by using the *contraflows* procedure, which increases the capacity of the road network by reversing the direction of certain lanes on major arteries [9]. In this way, contraflows can virtually double the capacity of the selected roads without affecting traffic safety [10].

In this paper we present, for the first time, a column-generation (CG) algorithm for zone-based evacuation planning that simultaneously produces convergent paths and non-preemptive schedules. Our algorithm can be easily extended to consider contraflows. We test our algorithm in a case study inspired by a real evacuation scenario and compare our approach to existing models in the literature that create convergent or non-preemptive plans, separately. Furthermore, we use a traffic simulator to evaluate the quality of our plans in real settings.

The remainder of this paper is organized as follows. Section 2 presents a literature review on existing models for zone-based evacuation. Section 3 formally presents our problem description. Section 4 presents our CG algorithm. Section 5 presents our case study and results. Finally, Section 6 concludes the paper and outlines research currently underway.

URI: https://hdl.handle.net/10125/79640
978-0-9981331-5-7

2. Literature Review

Evacuation planning models can be classified according to their precision level at a macroscopic or microscopic scale [11, 12]. Models at the macroscopic scale usually focus on strategic decisions, consider evacuees as homogeneous, and use network flows models in graphs expanded over discretized time [11]. For example, Lim et al. [13] evacuate three regions in the Greater Houston area in Texas using a capacity constrained network flow model on a time-expanded network. Models at a microscopic scale commonly use simulation models to capture the movement, behavior, and interactions of evacuees [12]. Pel et al. [14] presented a review on traffic simulation models used for evacuation planning. Models at the less-known mesoscopic scale usually couple optimization and simulation models in iterative feedback loops [15]. In this paper, we follow the macroscopic trend of using a time-expanded network in our optimization model; and we use a traffic simulator to microscopically evaluate the resulting evacuation plans.

The *zone-based evacuation planning problem* (ZEPP), firstly introduced by Pillac et al. [16], aims to find a single evacuation path for every zone at a macroscopic scale. To solve it, Pillac et al. [10] proposed a conflict-based path generation (CPG) heuristic that produces evacuation plans in a small computational time, with neither convergent paths nor non-preemptive schedules.

The problem of finding convergent paths for the ZEPP (C-ZEPP) was solved by Romanski and Van Hentenryck [17] using a Benders decomposition (Benders Convergent) algorithm. Their restricted master problem (RMP) is based on a tree-design problem (TDP) that maximizes the number of evacuated vehicles in a predefined time horizon while producing convergent paths in a network with aggregated capacities over time. The subproblem is a flow scheduling problem (FSP) that maximizes the number of evacuated vehicles and schedules their flow along the convergent paths (generated by the RMP) while ensuring the network capacity at any time. They used Pareto-optimal cuts to speed up the convergence of the algorithm, which stops when both the RMP and FSP evacuate the same number of vehicles. Furthermore, they embedded this Benders Convergent (B-C) algorithm within a binary search that finds the minimum clearance time of the evacuation. Although the B-C algorithm generates convergent paths efficiently, it does not produce non-preemptive schedules.

Even et al. [18] and Artigues et al. [19] find non-preemptive schedules for an evacuation using constraint programming (CP). Their models receive as input parameters the unique evacuation routes for each zone and determine their evacuation start times and their departure rates. Other approaches formulate the problem as a resource-constrained project scheduling problem (RCPSP) [20]. For example, Artigues et al. [21] propose a RCPSP model using a static network where every arc is a resource, the evacuation of a zone is a job that requires these resources, and the exchange of resources between jobs is constrained by conditional time lags. As all of these approaches receive the evacuation routes as inputs and focus on scheduling evacuees, they are more concerned with operational decisions. In contrast, in this paper we deal with strategic or tactical decisions by jointly considering the design of the evacuation routes and the scheduling of evacuees.

The problem of jointly determining the routes and the non-preemptive schedules for the ZEPP (NP-ZEPP) was solved by Pillac et al. [22] and Hasan and Van Hentenryck [23] by introducing the concept of response curves to model the behavioral response of the individuals to evacuation orders [14]. A response curve models the departure rate of a zone once its evacuation starts. Thus, a *plan* for a given zone is determined by three components: an evacuation path, an evacuation start time, and a response curve for vehicles in that zone to follow. They presented a CG (CG-NP) algorithm for the NP-ZEPP in which columns are plans for zones and the RMP is a set partitioning problem that selects a plan for every zone while ensuring the network capacity. Hasan and Van Hentenryck [23] solve in parallel multiple pricing subproblems (PSPs) that generate feasible plans for every zone and every response curve. They do so by solving a shortest path problem with resource constraints (SPPRC) in every PSP using a hybrid approach that first computes k-shortest paths [24] until finding an elementary one or hitting a K threshold and switching to solve the CSP using a mixed-integer programming (MIP) model. Although this CG-NP algorithm produced evacuation plans with non-preemptive schedules, the resulting paths were not convergent.

The use of contraflows has been extensively studied in the literature. Kim et al. [25] demonstrated that the problem of selecting which arcs to be used in contraflow in an evacuation network is $\mathcal{NP}$-hard. Recently, Pyakurel [26] presented efficient algorithms for the quickest evacuation planning problem using contraflows and flow-dependent transit times. Hasan and Van Hentenryck [4, 8] presented a systematic review and comparison of all the existing models for the ZEPP in settings with and without contraflows, but none of the compared models produced convergent paths and non-preemptive schedules simultaneously.

In this paper, we present a CG algorithm for the ZEPP that considers contraflows and, for the first time, simultaneously produces convergent paths and non-preemptive schedules. Our RMP is a set partitioning problem that selects a plan for every zone and generates convergent paths while ensuring the network capacity. The additional convergent constraints in the RMP enforce the PSPs to produce elementary paths. Thus, every PSP is a SPPRC solved using the Pulse algorithm [27] that outperforms the hybrid approach proposed by Hasan and Van Hentenryck [23]. We compare the performance and the generated plans of the B-C, the CG-NP, and our algorithm from a macroscopic perspective. Finally, we use a traffic simulator to evaluate the quality of the generated plans.

3. Problem Description

Consider an example evacuation scenario in which 20 vehicles have to be evacuated from one evacuation zone (i.e., 0) and there are two available safe zones (i.e., A and B). To represent this scenario, let $\mathcal{G} = (\mathcal{N} = \mathcal{E} \cup \mathcal{T} \cup \mathcal{S}, \mathcal{A})$ be a *static graph* in which $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{S}$ are the sets of evacuation, transit, and safe nodes, that represent evacuation zones, intersections of the road network, and safe zones, respectively; and $\mathcal{A}$ is the set of arcs that represent segments of the road network. For every evacuation node $k \in \mathcal{E}$, let d_k be its demand, which represents the number of vehicles to be evacuated. For every arc $e \in \mathcal{A}$, let τ_e be its travel time and u_e its capacity (vehicles per unit time). Let $\mathcal{A}_c \subseteq \mathcal{A}$ be the subset of arcs that can be used in contraflow and let $\bar{e} \in \mathcal{A}_c$ be the unique arc that goes in the opposite direction of arc $e \in \mathcal{A}_c$. Figure 1 presents the static graph of this example. Arcs in $\mathcal{A}_c$ appear with their corresponding opposite next to them.

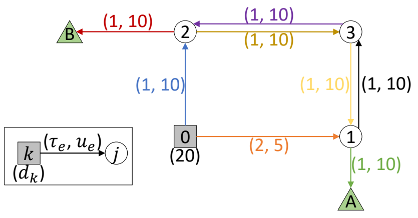

Figure 1. Static graph

To model traffic flow over time, we convert our static graph into a *time-expanded graph* $\mathcal{G}^x = (\mathcal{N}^x, \mathcal{A}^x)$. We first discretize time into periods of equal length. Let $\mathcal{H} = [0, h] \subset \mathbb{Z}$ be the set of periods in the planning horizon. Let $\Lambda(i) = \{ i_t \mid t \in \mathcal{H} \}$ be the set of *time-space copies* of node $i \in \mathcal{N}$ and let v be a *virtual sink node*. Thus, $\mathcal{N}^x = \cup_{i \in \mathcal{N}} \Lambda(i) \cup \{v\}$. For every static arc $e = (i, j) \in \mathcal{A}$ and every period $t \in \mathcal{H}$, let $e_t = (i_t, j_{t+\tau_e})$ be a *movement arc* that allows vehicles to flow through space and time. Each movement arc e_t is a time-space copy of e with capacity $u_{e_t} = u_e$. Let $\mathcal{M}_e = \{ e_t = (i_t, j_{t+\tau_e}) \mid t, t+\tau_e \in \mathcal{H} \}$ be the set of movement arcs generated by arc $e = (i, j) \in \mathcal{A}$. Thus, the set of all movement arcs is $\mathcal{A}^x_m = \cup_{e \in \mathcal{A}} \mathcal{M}_e$. Let $\mathcal{A}^x_w = \{ (k_t, k_{t+1}) \mid k \in \mathcal{E}, t \in \mathcal{H}, t < h \}$ be the set of *waiting arcs* that allow vehicles to wait until their departure. Let $\mathcal{A}^x_s = \{ (i_t, v) \mid i \in \mathcal{S}, t \in \mathcal{H} \}$ be the set of *sink arcs* that allow vehicles to reach the virtual sink node after arriving to a safe node. Waiting and sink arcs have infinite capacity. Thus, $\mathcal{A}^x = \mathcal{A}^x_m \cup \mathcal{A}^x_w \cup \mathcal{A}^x_s$. We further remove from the time-expanded graph every node and arc that cannot be reached from an evacuation node at period 0 or that cannot reach the virtual sink node. Finally, we let $n \in \mathcal{N}^x$ and $\alpha \in \mathcal{A}^x$ be a generic node and arc in $\mathcal{G}^x$; and $\delta^-(\cdot)$ and $\delta^+(\cdot)$ be the sets of incoming and outgoing arcs of *any* node in $\mathcal{N}$ or $\mathcal{N}^x$. Figure 2 presents the time-expanded graph in our example. Removed nodes and arcs are greyed out, every arc is labeled with its capacity, and all movement arcs $\mathcal{M}_e$ are in the same color as static arc $e \in \mathcal{A}$.

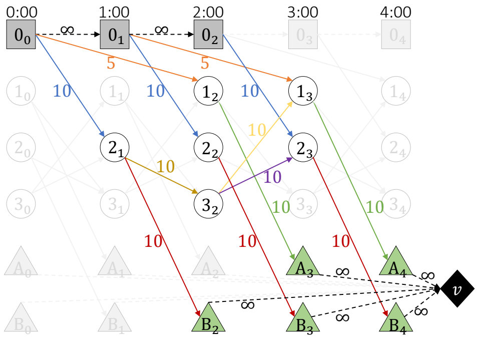

Figure 2. Time-expanded graph

Let $\mathcal{F}_k$ be the set of predefined response curves associated with evacuation node $k \in \mathcal{E}$ that model the number of departing vehicles from k at every period after its start period $t_0 \in \mathcal{H}$. For every $f \in \mathcal{F}_k$, let F be its cumulative function. Let $D_k(t)$ be the number of departing vehicles from evacuation node $k \in \mathcal{E}$ at period $t \in \mathcal{H}$ under response curve $f \in \mathcal{F}_k$. Formally,

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

Figure 3 shows the values of $D_k(t)$ under different response curves with an evacuation start time $t_0 = 60$.

Clearly, $D_k(t)$ can be used to define a non-preemptive schedule (i.e., uninterrupted departures).

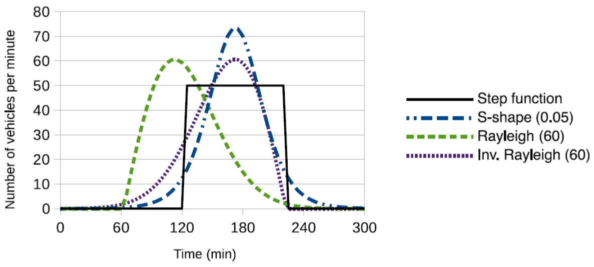

Figure 3. Number of departing vehicles with different response curves (from [22])

The following definitions are necessary to formally describe our problem.

Definition 1. A graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ is *connected* if for each $k \in \mathcal{E}$, there exists a path P from k to a safe node.

Definition 2. A graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ is *convergent* if for each $i \in \mathcal{E} \cup \mathcal{T}$, the outdegree of i is 1.

As stated by Even et al. [6], any connected evacuation graph $\mathcal{G}$ has 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 zone. Now we are ready to formally define our problem.

Definition 3. Given a connected evacuation graph $\mathcal{G}$, the Convergent and Non-preemptive Evacuation Planning Problem (CNP-ZEPP) consists on finding a convergent subgraph $\mathcal{G}' \subseteq \mathcal{G}$, an evacuation start time $t_0 \in \mathcal{H}$, and a response curve $f \in \mathcal{F}_k$ for every evacuation node $k \in \mathcal{E}$ that maximizes the number of evacuated vehicles in the minimum number of periods available and respects the network capacity.

Definition 4. The CNP-ZEPP with Contraflows (CNP-ZEPP-CF) has the addition of finding the subset $\mathcal{A}_c^* \subseteq \mathcal{A}_c$ of arcs to be used in contraflow.

4. The Column-Generation Algorithm

The key idea behind our CG algorithm for the CNP-ZEPP (CG-CNP), which we borrow from Pillac et al. [22], is to generate *time-response plans* for the evacuation nodes, henceforward referred to as *plans*. A plan for the evacuation node $k \in \mathcal{E}$ is of the form $p = \langle P, f, t_0 \rangle$, where P is an evacuation path in the static graph (it should start at k and end at a safe node) that vehicles in k should follow; $f \in \mathcal{F}_k$ is a response curve specifically adapted for k's demand from a predefined set $\mathcal{F}$ of response curves; and t_0 is the period at which the vehicles from k start evacuating following the response curve f.

Our CG-CNP is composed by a restricted master problem (RMP) and multiple Pricing Subproblems (PSPs). The RMP is the linear relaxation of an integer program (IP) that selects plans for every evacuation node from a small subset of available generated plans, while ensuring network capacity. Each PSP, one for every evacuation node and every response curve, generates feasible promising plans that improve the incumbent solution. The algorithm has two phases: a column-generation phase and a last iteration. At every iteration of the column-generation phase, we first solve the RMP and then the PSPs until no other promising plan is generated. Then, in the last iteration, we solve the RMP as an IP with the generated columns. Thus, our algorithm is a heuristic for the CNP-ZEPP. We refer the interested reader to [28, 29] for a review of techniques and applications of CG on MIPs.

4.1. The Restricted Master Problem

The RMP is a set partitioning problem that selects plans from a subset of generated plans while ensuring that the selected plans induce a convergent graph and respect its capacity. Its objective is a multi-criteria function that aims to minimize the evacuation time and maximize the number of evacuated vehicles.

Let Ω'_k be the subset of generated plans for evacuation node $k \in \mathcal{E}$, initially populated with an empty plan that evacuates no vehicles. Thus, $\Omega' = \cup_{k \in \mathcal{E}} \Omega'_k$ is the subset of all generated plans. Let $\omega(e) \subset \Omega'$ be the subset of generated plans whose path contains arc $e \in \mathcal{A}$, i.e., $\omega(e) = \{p \in \Omega' \mid e \in p\}$. Let a_{p,e_t} be the flow of vehicles on arc $e \in \mathcal{A}$ at period $t \in \mathcal{H}$ induced by plan $p \in \Omega'$. Let c_p be the cost of plan $p \in \Omega'_k$, which considers the time at which vehicles arrive to a safe zone and heavily penalizes the number of non-evacuated vehicles. Formally, c_p is defined as

$$c_p = \sum_{e \in p} \sum_{t \in \mathcal{H}} c_{e_t} \cdot a_{p,e_{e_t}} + \bar{c} \cdot \bar{d}_p, \tag{2}$$

where c_{e_t}, defined by Equation (3), is a cost for movement arc $e_t \in \mathcal{A}_m^x$ that tallies the normalized time at which a vehicle arrives in a safe node; $\bar{c}$, defined by Equation (4), is a heavy penalty for every non evacuated vehicle; and $\bar{d}_p$ is the number of vehicles from $k \in \mathcal{E}$ that could not be evacuated using plan $p \in \Omega'_k$.

$$c_{e_t} = c_{(i,j)_t} = \begin{cases} \frac{t}{h}, & \text{if } j \in \mathcal{S}; \\ 0, & \text{otherwise;} \end{cases} \tag{3}$$

$$\bar{c} = 100 \cdot \max_{e_t \in \mathcal{A}_m^x} \{c_{e_t}\} \cdot \max_{k \in \mathcal{E}} \{d_k\}; \tag{4}$$

The RMP considers a binary variable y_p that indicates whether plan $p \in \Omega'$ is selected and a binary variable x_e that indicates if arc $e \in \mathcal{A}$ is included in the convergent graph. The mathematical formulation of the RMP is as follows:

$$\text{minimize} \sum_{p \in \Omega'} c_p \cdot y_p \tag{5}$$

Subject to,

$$\sum_{p \in \Omega'_k} y_p = 1, \qquad \forall\, k \in \mathcal{E}; \tag{6}$$

$$\sum_{e \in \delta^+(i)} x_e \leq 1, \qquad \forall\, i \in \mathcal{T}; \tag{7}$$

$$\sum_{p \in \omega(e)} y_p \leq |\omega(e)| \cdot x_e, \qquad \forall\, e \in \mathcal{A}; \tag{8}$$

$$\sum_{p \in \omega(e)} a_{p,e_t} \cdot y_p \leq u_{e_t}, \qquad \forall\, e_t \in \mathcal{A}^x_m; \tag{9}$$

$$y_p \in [0,1], \qquad \forall\, p \in \Omega'; \tag{10}$$

$$x_e \in [0,1], \qquad \forall\, e \in \mathcal{A}. \tag{11}$$

Objective function (5) is a multi-criteria function that jointly minimizes the overall time of the evacuation and maximizes the number of evacuated vehicles. Constraints (6) guarantee that a plan is selected for every evacuation node. Constraints (7) guarantee that the resulting evacuation graph is convergent (see Definition 2). Constraints (8) prevent from selecting a plan that uses an arc not present in the convergent graph. Constraints (9) preserve the capacity of the network. Constraints (10)-(11) define the variables' domain.

After solving the RMP at every iteration of the column-generation phase, we retrieve the dual variables $\{\pi_k\}$, $\{\pi_e\}$, and $\{\pi_{e_t}\}$ of constraints (6), (8), and (9), respectively and plug them into the PSPs. Finally, in the last iteration, we set the variables to binary.

4.2. The Pricing Subproblem

The PSP is in charge of finding promising plans to include in the RMP. Since each plan is independent to the others, multiple PSPs, one for every evacuation node $k \in \mathcal{E}$ and every response curve $f \in \mathcal{F}_k$, can be solved concurrently in parallel to find multiple promising plans. Therefore, henceforward in this subsection, when we refer to a promising plan p, we imply that p is for evacuation node $k \in \mathcal{E}$ and uses response curve $f \in \mathcal{F}_k$.

A promising plan has a negative reduced cost in the RMP, implying that the new plan will decrease the objective function if it enters the RMP basis. Thus, the PSP minimizes the reduced cost r_p of the generated plan. Namely,

$$r_p = \bar{c}\cdot\bar{d}_p - \pi_k - \sum_{e \in p} \pi_e + \sum_{e_t \in \mathcal{A}^x_m} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t}. \tag{12}$$

Let us gain insight on the above expression. First, note that the only term that is independent to the plan's path is π_k. Thus, to find a plan with negative reduced cost, we can find a path P and a start time t_0 that minimizes the other terms in Equation (12). Let these terms be grouped as:

$$C^k_f(P, t_0) = \bar{c} \cdot \bar{d}_p - \sum_{e \in p} \pi_e + \sum_{e_t \in \mathcal{A}^x_m} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t}. \tag{13}$$

Second, consider the highlighted time-expanded path $\{\, 0_0, 0_1, 2_2, B_3, v \,\}$ in Figure 4. Its projected path in the static graph is $\{\, 0, 2, B \,\}$, which departs from 0 at period 1. Let the generated plan use this (static) path and this start period. Then, vehicles will also use the other non-highlighted (time-expanded) path in Figure 4. Since the generated plan's response curve is known *a priori*, the highlighted time-expanded path uniquely determines the flow of vehicles in advance. Thus, flows can be aggregated in advance.

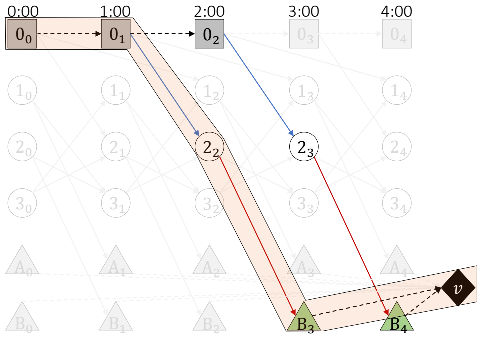

Figure 4. Path in the time-expanded graph

Therefore, to jointly find a path P (in the static graph) and an evacuation start period t_0 for a new plan (for evacuation node $k \in \mathcal{E}$), we can just find path P^x (in the time-expanded graph) from k_0 to v. P is the projected path of P^x into the static network and t_0 is the period at which P^x departs from a time-space node $\Lambda(k)$ using any movement arc.

To find a path P^x that minimizes $C^k_f(P, t_0)$, it is necessary to define the following costs:

$$c^{\text{sp}}_{\alpha} = 0, \qquad \forall\, \alpha \in \mathcal{A}^x_w. \tag{14}$$

$$c^{\text{sp}}_{\alpha} = c^{\text{sp}}_{(i_t,v)} = \bar{c} \cdot (d_k - F(h-t)), \qquad \forall\, \alpha \in \mathcal{A}^x_s; \tag{15}$$

$$c^{\text{sp}}_{e_t} = -\pi_e + \sum_{t'=t}^{h} (c_{e_{t'}} - \pi_{e_{t'}}) \cdot f(t'-t), \quad \forall\, e_t \in \mathcal{A}^x_m; \tag{16}$$

Equation (15) accounts for the penalty cost of non-evacuated vehicles with the generated plan, which is the first term in Equation (13). Since flows can be aggregated and the accumulated function F of the response curve is known *a priori*, the number of non-evacuated vehicles can be computed and thus, its penalty. Equation (16) aggregates future costs of every movement arc. However, there is a crucial consideration to note. The summation in Equation (16) accounts for the last summation in Equation (13) by aggregating future costs of the movement arcs. Nonetheless, note that in Equation (13), there is one term π_e for every arc $e \in p$. However, note that all the movement arcs $\mathcal{M}_e$ have the same term π_e in Equation (16). It follows that the resulting path P^x cannot have more than one movement arc $\mathcal{M}_e$ of any $e \in \mathcal{A}$ and hence

$$\begin{aligned}
\sum_{\alpha \in P^x} c^{\text{sp}}_{\alpha} = &\sum_{(i_t,v) \in \mathcal{A}^x_s \cap P^x} \bar{c} \cdot (d_k - F(h-t)) - \sum_{e_t \in \mathcal{A}^x_m \cap P^x} \pi_e + \\
&\sum_{e_t \in \mathcal{A}^x_m \cap P^x} \sum_{t'=t}^{h} (c_{e_{t'}} - \pi_{e_{t'}}) \cdot f(t'-t) \\
&= \bar{c} \cdot \bar{d}_p - \sum_{e \in p} \pi_e + \sum_{e_t \in \mathcal{A}^x_m} (c_{e_t} - \pi_{e_t}) \cdot a_{p,e_t} \\
&= C^k_f(P, t_0)
\end{aligned} \tag{17}$$

Thus, to find a path P^x that minimizes $C^k_f(P,t_0)$, we can find a shortest path in the time-expanded network, using cost c^{sp}_{α} for every arc $\alpha \in \mathcal{A}^x$, and guaranteeing that no more than one movement arc $\mathcal{M}_e$ is used for any static arc $e \in \mathcal{A}$. Note that this last constraint is equivalent to P^x not having more than one time-space node $\Lambda(i)$ for any transit node $i \in \mathcal{T}$. Thence, each PSP is a SPPRC that ultimately produces elementary paths in the static graph as in [23].

Let x_α be a binary variable that indicates whether arc $\alpha \in \mathcal{A}^x$ is used in path P^x. The mathematical formulation of each PSP is as follows:

$$\text{minimize} \sum_{\alpha \in \mathcal{A}^x} c^{\text{sp}}_{\alpha} \cdot x_{\alpha} \tag{18}$$

Subject to,

$$\sum_{\alpha \in \delta^+(n)} x_\alpha - \sum_{\alpha \in \delta^-(n)} x_\alpha = \begin{cases} 1, & \text{if } n = k_0; \\ 0, & \forall\, n \in \mathcal{N}^x \setminus \{k_0, v\}; \\ -1, & \text{if } n = v; \end{cases} \tag{19}$$

$$\sum_{i_t \in \Lambda(i)} \sum_{e_t \in \delta^+(i_t) \cap \mathcal{A}^x_m} x_{e_t} \le 1, \qquad \forall\, i \in \mathcal{T}; \tag{20}$$

$$x_\alpha \in \{0,1\}, \quad \forall\, \alpha \in \mathcal{A}^x. \tag{21}$$

Objective function (18) minimizes the cost of P^x. Constraints (19) are the classical flow-balance constraints. Constraints (20) are the resource constraints. Finally, constraints (21) define the variables' domain.

Every PSP tries to find a plan p for evacuation node $k \in \mathcal{E}$ using response curve $f \in \mathcal{F}_k$. After finding the constrained shortest path P^x, we can find all the remaining components of p. The path P can be obtained by projecting P^x into the static network. The plan's start time t_0 is the time at which P^x leaves a node $\Lambda(k)$ using a movement arc. Formally,

$$t_0 = \sum_{k_t \in \Lambda(k)} \sum_{e_t \in \delta^-(k_t)} t \cdot x_{e_t} \tag{22}$$

By substituting Equations (17) and (13) into Equation (12), the reduced cost r_p of the generated plan is

$$r_p = \sum_{\alpha \in \mathcal{A}^x} c^{\text{sp}}_{\alpha} \cdot x_\alpha - \pi_k \tag{23}$$

Finally, if the reduced cost of the generated plan p satisfies $r_p < 0$ and $p \notin \Omega'_k$, we include p in the subset of generated plans Ω'_k and, consequently, in Ω'. When no additional plan is generated, the column-generation phase ends and the last iteration solves the RMP as an IP with the generated plans.

The result of the algorithm is the set $\Omega'^* \subset \Omega'$ of the selected plans, one for every evacuation node, which induce a convergent subgraph $\mathcal{G}^* = (\mathcal{N}^*, \mathcal{A}^*)$.

4.3. The Contraflow Extension

Our CG-CNP can be easily extended to consider contraflows by following the same intuition that Hasan and Van Hentenryck [4] used to consider contraflows in the B-C algorithm proposed by Romanski and Van Hentenryck [17]. Because of Constraints (7), which guarantee that the outdegree of every node is at most 1, it follows that arc $e \in \mathcal{A}_c$ and its opposite $\bar{e} \in \mathcal{A}_c$ cannot be in the resulting convergent graph $\mathcal{G}^*$ at the same time. Therefore, if $e \in \mathcal{A}_c$ is used in the resulting convergent graph (i.e., $e \in \mathcal{A}^*$), its opposite $\bar{e}$ can be safely used in contraflow as it is guranteed that $\bar{e}$ is not used by any other evacuation path (i.e, $\bar{e} \notin \mathcal{A}^*$).

Therefore, to consider contraflows, let u^o_e be the original capacity of arc $e \in \mathcal{A}_c$. Before constructing the time-expanded graph, we set the capacity of every arc $e \in \mathcal{A}_c$ to $u_e = u^o_e + u^o_{\bar{e}}$. Then, we proceed to build the time-expanded graph with the augmented capacities and execute the CG-CNP algorithm as usual. After its completion, we can easily determine whether an arc is used in contraflow by checking if the flow on its opposite arc exceeds its original capacity at any period.

$$\mathcal{A}^*_c = \left\{ e \in \mathcal{A}_c \;\middle|\; \exists\, t \in \mathcal{H} \text{ s.t. } \sum_{p \in \Omega'^*} a_{p,\bar{e}_t} > \tau_{\bar{e}} \cdot u^o_{\bar{e}} \right\} \tag{24}$$

4.4. The Solution Approach for the PSP

When solving the RMP in the last iteration (as an IP), if Ω' contains all the possible existing plans for every zone, then it is guaranteed that the optimal solution will be found. However, in practice, there are available only a small subset of generated plans. Having a larger set of plans increments the chances of finding a better solution in the last iteration. Nonetheless, to get a plan, every PSP must solve a SPPRC, which is $\mathcal{NP}$-hard [27]. Hasan and Van Hentenryck [23] solved the SPPRC in every PSP using a Hybrid algorithm that has two stages. The first stage attempts to solve the SPPRC indirectly using the Recursive Enumeration Algorithm (REA) [24] to iteratively find k-shortest paths. The REA stops as soon as it finds a path P^x that, when projected into the static graph, results in an elementary path P. However in practice, many instances required a large k, consuming limited memory and time. For that reason, they impose a K limit that, when reached, switches the Hybrid algorithm to the second stage that solves the SPPRC directly using a MIP solver. Therefore, the Hybrid algorithm can generate at most one plan per PSP.

In this paper, we use the *Pulse* algorithm by Lozano and Medaglia [27] to directly solve the SPPRC in every PSP. The Pulse is an specialized algorithm for the SPPRC that has been successfully extended and used to solve different problems [30]. The key idea is to recursively propagate *pulses* from a source node towards an end node. As a pulse traverses the network from node to node, it builds a partial path P^x and stores multiple attributes associated with it, such as the cumulative cost or resource consumption. At the core of the algorithm, the *pruning strategies* by *dominance*, *feasibility*, and *bounds* prevent the propagation of a pulse as soon as there is enough evidence that the partial path will not lead to a feasible or improved solution. Then, each pulse that reaches the final node contains all the information for a feasible path that is associated with a new plan. This plan is included in the subset of generated plans if it satisfies the reduced cost criterion. Hence, the key advantage of the Pulse over the Hybrid approach is its ability to generate multiple plans per PSP.

In our simplest version of the Pulse, we propagate pulses in the time-expanded network in a *forward* direction from the evacuation node k_0 in each PSP towards the virtual sink node v. The resources are associated with each one of the transit nodes $\mathcal{T}$. Therefore, as soon as a pulse visits more than one node in $\Lambda(i)$ for any $i \in \mathcal{T}$, we safely prune it because it has already consumed all the resource available. Additionally, for the pruning by dominance, we use an elitist rule by fixing one label in every node that stores the best cost among the paths that have visited them. Finally, we use the original prune by bounds.

One of our contributions is the flexibility when using the Pulse to discover even more plans per PSP. In case that more plans are needed for any reason, e.g., not all the evacuees could be evacuated, we can easily extend the Pulse to find more plans that could resolve this issue. To do so, we use the *Pulse F+B*: After running the forward Pulse in every PSP, we run a *backwards* Pulse from the virtual sink node v towards the evacuation node k_0. The Pulse F+B allows us to find new paths that may not be found in the forward direction, at the expense of more computational time, and with the potential benefit of finding new plans that could remedy any lack of options to completely evacuate the desired region. Experimental results presented in Section 5 demonstrate the benefits of this approach.

5. The Case Study

Our case study is the evacuation of 38,343 vehicles in the Hawkesbury-Nepean (HN) floodplain, located at the north-west of Sydney, Australia. The resulting static graph has 80 evacuation nodes, 184 transit nodes, 5 safe nodes, and 580 edges. Our planning horizon is 10 hours with 5-minute periods, resulting in $h = 120$. We first present a comparison between the Hybrid and the Pulse algorithms for solving the PSPs in the CG-CNP. Then, we present a macroscopic evaluation of our algorithm and other algorithms in the literature for the ZEPP. Finally, we present a microscopic evaluation of the same algorithms using a traffic simulator. All of the algorithms were implemented in JAVA 8 and solved with GUROBI 9.2. All of our experiments were ran using a PC with an Intel® Core^{TM} i7-1085H processor running at 2.7GHz, 32 GB RAM, and using Windows 10 Pro.

5.1. The Hybrid vs. Pulse Comparison

We ran the CG-CNP algorithm specifying a set of step response curves $\mathcal{F}$ with departure rates of 2, 5, 10, 25, and 50 vehicles per period. Additionally, we imposed a limit of 24 hours for the last iteration and a limit of 48 hours for all the algorithm. We set a K limit of 10^5 paths for the REA and a time limit of 300 seconds in the MIP for the SPPRC in the Hybrid algorithm.

Table 1 presents the results of the CG-CNP. We first focus our attention on the rows associated with the Pulse and the Hybrid approaches in both settings without and with contraflows. We can evidence that the Pulse generates more plans, which allow the column-generation phase to converge in fewer iterations and in less computational time. Figure 5 presents a clear picture of how drastically the Pulse outperforms the Hybrid algorithm when solving all the PSPs in every iteration of the column-generation phase (in the scenario without contraflows). The average speedup achieved is up to 47.8 when the Pulse is faster and 20.2 in general. Furthermore, the plans generated allow the last iteration to converge even faster to a final solution (Column 7). Overall, the Pulse always outperforms the Hybrid algorithm in terms of CPU time. This is specially relevant considering that the CG-CNP is a heuristic algorithm for the CNP-ZEPP.

Now, we focus our attention on the Pulse F+B. Overall, the additional plans found with the backwards Pulse heavily impact the time to solve the RMP in the last iteration of the algorithm. However, this approach is the one that produces the best objective functions. In the setting with contraflows, the new experiment also evacuates everyone within the same evacuation time as the forward Pulse. In contrast, without contraflows, neither the Pulse nor the Hybrid approaches can evacuate all the vehicles. However, with the additional plans found with the Pulse F+B, the algorithm is not only able to evacuate all the vehicles but it does more efficiently (i.e., 585 mins). A possible rationale for this behavior is that the backwards Pulse, due to its nature, discovers new paths from the future to the past that could not be found using the forward Pulse.

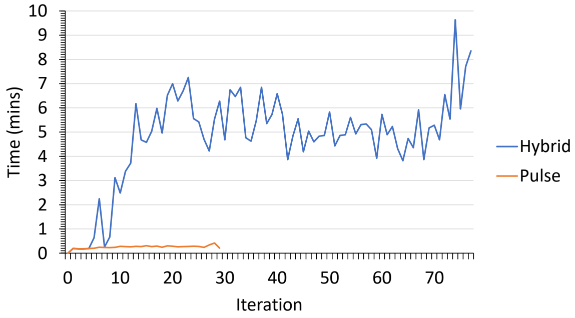

Figure 5. Time (mins) solving the PSPs

5.2. The Macroscopic Comparison

We now compare our CG-CNP algorithm for the CNP-ZEPP against the B-C for the C-ZEPP [17], and the CG-NP for the NP-ZEPP [23]. We use the results of the B-C algorithm for minimizing the clearance time and imposing a time limit of 300 seconds on the FSP. For the CG-NP, we used the same parameters as the CG-CNP. Table 2 presents the results of all the algorithms. Among the three versions of the ZEPP, the C-ZEPP is the most relaxed one because it allows evacuation schedules without conservative departure rates that can be interrupted anytime. On the other hand, the CNP-ZEPP is the most constrained version because it requires paths to be convergent and schedules to be non-preemptive with conservative departure rates. Both the B-C and the CG-NP, which solve more relaxed versions of the ZEPP, are able to evacuate 100% of the vehicles. However, the CG-CNP with neither the Hybrid nor the Pulse, which solve the most constrained version of the ZEPP, can evacuate them all. Columns 6 and 9 clearly show that as the problem becomes more constrained, the evacuation takes more time.

An interesting insight from Columns 4 and 7 is that the convergent constraints in the RMP allow the CG-CNP converge faster than the CG-NP. Moreover, considering contraflows relaxes the capacity constraints and allows all the CG algorithms to find a final solution even faster. Then, convergent plans are inherently amenable to contraflows.

5.3. The Microscopic Evaluation

None of the above models capture microscopic attributes that ultimately affect the evacuation dynamics. For this reason, similarly to Hasan and Van Hentenryck [8], we use Sumo (Simulation for Urban Mobility) traffic simulation package [31] to evaluate the performance of the generated evacuation plans in a more realistic scenario. We use actual

Table 1. Results of the CG-CNP algorithm for the CNP-ZEPP

Contraflows	algorithm used in PSPs	Column-generation phase				Last iteration		Algorithm		
		Generated plans	Number of iterations	CPU time (mins)	Final objective function value	CPU time (mins)	Final objective function value	CPU time (mins)	Evacuated percentage	Evacuation time (mins)
No	Hybrid	11,312	77	408.43	8,815.61	291.03	20,857,053.13	699.46	99.14	590
	Pulse	17,284	30	104.82	8,815.61	43.45	36,995,819.98	148.28	98.72	590
	Pulse F+B	40,567	20	514.96	8,815.61	1,440.02	13,731.27	1,954.98	100.00	585
Yes	Hybrid	3,023	49	14.13	6,202.69	1.19	8,631.05	15.34	100.00	470
	Pulse	3,517	28	5.91	6,202.69	2.41	8,536.44	8.36	100.00	455
	Pulse F+B	5,834	20	4.86	6,202.69	21.43	8,312.58	26.31	100.00	455

Table 2. Macroscopic results

Problem	Algorithm	Version	Without contraflows			With contraflows		
			CPU time (mins)	Evacuated percentage	Evacuation time (mins)	CPU time (mins)	Evacuated percentage	Evacuation time (mins)
C-ZEPP	B-C	Min. clearance	0.26	100.00	335	1.10	100	210
NP-ZEPP	CG-NP	Hybrid	2,881.40	100.00	480	1,495.72	100	245
CNP-ZEPP	CG-CNP	Hybrid	699.46	99.14	590	15.34	100	470
		Pulse	148.28	98.72	590	8.36	100	455
		Pulse F+B	1,954.98	100.00	585	26.31	100	455

data of the road network such as speed limits, lane counts, and GPS coordinates to construct a simulation that considers vehicles' instantaneous speed, safe distances between vehicles, car-following behavior, and lane-changing behavior. The path and departure times for the evacuated vehicles are retrieved from the solutions of the algorithms exactly as generated.

Table 3 presents the results of the single-run simulation of the five algorithms' solutions without contraflows. Note that the evacuated percentages achieved with the five simulations are equal to the macroscopic results. However, the evacuation times are different. For this reason, we calculated the evacuation time ratio as the fraction between the microscopic (numerator) and the macroscopic (denominator) evacuation times. A ratio closer to 1 indicates that the gap between the optimization and the simulation times is lower, which makes the optimization solution more realistic.

In the B-C algorithm, the generated schedule specifies departure rates that could induce the utilization of the road network at full capacity. Thus, in the simulation, the road network gets saturated, producing congestion hot spots that severely delay the evacuation. So the simulation time becomes greater than the optimization time, causing a ratio above 1. However, with the CG algorithms, the non-preemptive schedules with conservative departure rates prevent the road network to be used at its full capacity. Thus, in the simulation, the produced delay due to congestion is so small that it cannot overcome the overestimation of the transit times due to time discretization. Thus, the simulation times becomes lesser than the optimization times, so ratios are below 1. Overall, convergent paths and non-preemptive schedules with conservative rates allow the generated plans to be more realistic and behave accordingly in real-world scenarios.

Table 3. Microscopic results without contraflows

Simulated solution		Evacuated percentage	Evacuation time (mins)	Evacuation time ratio
Algorithm	Version			
B-C	Min. clearance	100.00	398.82	1.19
CG-NP	Hybrid	100.00	402.18	0.84
CG-CNP	Hybrid	99.14	562.47	0.95
	Pulse	98.72	567.73	0.96
	Pulse F+B	100.00	544.12	0.93

6. Concluding Remarks

In this paper we presented a column-generation algorithm for zone-based evacuation planing that, for the first time, jointly produces convergent evacuation paths, determines non-preemptive schedules, and selects roads to be used in contraflow. The key idea of the algorithm is to use time-response curves with conservative departure rates that prevent the road network to be used at its full capacity. We embedded the Pulse algorithm within our CG to find high-quality solutions efficiently. We compared our results against existing models in the literature that produce separately convergent paths or non-preemptive schedules. When evaluated under the lens of a microscopic traffic simulation, the joint use of convergent paths and non-preemptive schedules produce plans closer to real-world settings. Overall, our algorithm produces high-fidelity actionable evacuation plans.

Future research will integrate the design of routes into the scheduling models in [18, 21], and using instance generators in the literature [32] to compare different approaches. Research currently underway includes considering electric vehicles (EVs) in the evacuation in such a way that the electric power grid does not get overloaded when charging the EVs.

7. Acknowledgments

The authors would like to thank Mohd. Hafiz Hasan for his thorough explanations, companion, and revisions.

References

[1] I. H. Sawalha, "A contemporary perspective on the disaster management cycle," *foresight*, vol. 22, pp. 469–482, Jan 2020.

[2] B. Wolshon, E. Urbina, C. Wilmot, and M. Levitan, "Review of policies and practices for hurricane evacuation. i: Transportation planning, preparedness, and response," *Natural Hazards Review*, vol. 6, no. 3, pp. 129–142, 2005.

[3] SES-NSW, "Hawkesbury nepean flood emergency sub plan," tech. rep., State Emergency Service - New South Whales, 2005.

[4] M. H. Hasan and P. Van Hentenryck, "Large-scale zone-based evacuation planning—part i: Models and algorithms," *Networks*, vol. 77, no. 1, pp. 127–145, 2021.

[5] F. F. Townsend, "The federal response to hurricane katrina: Lessons learned," tech. rep., The White House, Washington, DC, Feb 2006.

[6] C. Even, V. Pillac, and P. Van Hentenryck, "Convergent plans for large-scale evacuations," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, p. 1121–1127, AAAI Press, 2015.

[7] J. A. Huertas, D. Duque, E. Segura-Durán, R. Akhavan-Tabatabaei, and A. L. Medaglia, "Evacuation dynamics: a modeling and visualization framework," *OR Spectrum*, vol. 42, no. 3, pp. 661–691, 2020.

[8] M. H. Hasan and P. Van Hentenryck, "Large-scale zone-based evacuation planning, part ii: Macroscopic and microscopic evaluations," *Networks*, vol. 77, no. 2, pp. 341–358, 2021.

[9] G. Theodoulou and B. Wolshon, "Alternative methods to increase the effectiveness of freeway contraflow evacuation," *Transportation Research Record*, vol. 1865, no. 1, pp. 48–56, 2004.

[10] V. Pillac, P. Van Hentenryck, and C. Even, "A conflict-based path-generation heuristic for evacuation planning," *Transportation Research Part B: Methodological*, vol. 83, pp. 136–150, 2016.

[11] H. Hamacher and S. Tjandra, "Mathematical modelling of evacuation problems: A state of art," Tech. Rep. 24, Fraunhofer (ITWM), 2001.

[12] V. Bayram, "Optimization models for large scale network evacuation planning and management: A literature review," *Surveys in Operations Research and Management Science*, vol. 21, no. 2, pp. 63–84, 2016.

[13] G. J. Lim, S. Zangeneh, M. Reza Baharnemati, and T. Assavapokee, "A capacitated network flow optimization approach for short notice evacuation planning," *European Journal of Operational Research*, vol. 223, no. 1, pp. 234–245, 2012.

[14] A. J. Pel, M. C. J. Bliemer, and S. P. Hoogendoorn, "A review on travel behaviour modelling in dynamic traffic simulation models for evacuations," *Transportation*, vol. 39, pp. 97–123, Jan 2012.

[15] J. A. Zambrano, J. A. Huertas, E. Segura-Durán, and A. L. Medaglia, "Time estimation and hotspot detection in the evacuation of a complex of buildings: A mesoscopic approach and case study," *IEEE Transactions on Engineering Management*, vol. 67, no. 3, pp. 641–658, 2020.

[16] V. Pillac, P. Van Hentenryck, and C. Even, "A conflict-based path-generation heuristic for evacuation planning," *CoRR*, vol. abs/1309.2693, 2013.

[17] J. Romanski and P. Van Hentenryck, "Benders decomposition for large-scale prescriptive evacuations," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 3894–3900, AAAI Press, 2016.

[18] C. Even, A. Schutt, and P. Van Hentenryck, "A constraint programming approach for non-preemptive evacuation scheduling," in *Principles and Practice of Constraint Programming* (G. Pesant, ed.), (Cham), pp. 574–591, Springer International Publishing, 2015.

[19] C. Artigues, E. Hébrard, Y. Pencolé, A. Schutt, and P. J. Stuckey, "A Study of Evacuation Planning for Wildfires," in *The Seventeenth International Workshop on Constraint Modelling and Reformulation (ModRef 2018)*, (Lille, France), p. 17p., Aug. 2018.

[20] H. Toussaint and A. Quilliot, "Flow Polyhedra and Resource Constrained Project Scheduling Problems," *RAIRO - Operations Research*, vol. 46, pp. 373 – 409, Nov. 2012.

[21] C. Artigues, A. Quilliot, H. Toussaint, and E. Hebrard, "Models and algorithms for natural disaster evacuation problems," in *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 143–146, 2019.

[22] V. Pillac, M. Cebrian, and P. Van Hentenryck, "A column-generation approach for joint mobilization and evacuation planning," *Constraints*, vol. 20, pp. 285–303, Jul 2015.

[23] M. H. Hasan and P. Van Hentenryck, "A column-generation algorithm for evacuation planning with elementary paths," in *Principles and Practice of Constraint Programming* (J. C. Beck, ed.), (Cham), pp. 549–564, Springer International Publishing, 2017.

[24] V. M. Jiménez and A. Marzal, "Computing the k shortest paths: A new algorithm and an experimental comparison," in *Algorithm Engineering* (J. S. Vitter and C. D. Zaroliagis, eds.), (Berlin, Heidelberg), pp. 15–29, Springer Berlin Heidelberg, 1999.

[25] S. Kim, S. Shekhar, and M. Min, "Contraflow transportation network reconfiguration for evacuation route planning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1115–1129, 2008.

[26] U. Pyakurel, H. N. Nath, and T. N. Dhamala, "Efficient contraflow algorithms for quickest evacuation planning," *Science China Mathematics*, vol. 61, pp. 2079–2100, Nov 2018.

[27] L. Lozano and A. L. Medaglia, "On an exact method for the constrained shortest path problem," *Computers & Operations Research*, vol. 40, no. 1, pp. 378–384, 2013.

[28] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column generation*. Springer, 2005.

[29] M. E. Lübbecke and J. Desrosiers, "Selected Topics in Column Generation," *Operations Research*, vol. 53, pp. 1007–1023, dec 2005.

[30] A. Medaglia, L. Lozano, and D. Duque, "Solving hard shortest path problems with the pulse framework," *IFORS News. Tutorial Section.*, vol. 12, 06 2018.

[31] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[32] C. Artigues, E. Hébrard, Y. Pencolé, A. Schutt, and P. J. Stuckey, "Data Instance generator and optimization models for evacuation planning in the event of wildfire," in *GEOSAFE Workshop on Robust Solutions for Fire Fighting (RSFF 2018)* (A. N. Gabriele Di Stefano, ed.), vol. 2146 of *Robust Solutions for Fire Fighting*, (L'Aquila, Italy), pp. 75–86, CEUR-WS, July 2018.