Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Convergent Plans for Large-Scale Evacuations

Caroline Even[1] and **Victor Pillac**[1] and **Pascal Van Hentenryck**[1,2]
[1]National ICT Australia (NICTA), Melbourne, Australia
[2]Australian National University (ANU), Canberra, Australia

Abstract

Evacuation planning is a critical aspect of disaster preparedness and response to minimize the number of people exposed to a threat. Controlled evacuations aim at managing the flow of evacuees as efficiently as possible and have been shown to produce significant benefits compared to self-evacuations. However, existing approaches do not capture the delays introduced by diverging and crossing evacuation routes, although evidence from actual evacuations highlights that these can lead to significant congestion. This paper introduces the concept of convergent evacuation plans to tackle this issue. It presents a MIP model to obtain optimal convergent evacuation plans which, unfortunately, does not scale to realistic instances. The paper then proposes a two-stage approach that separates the route design and the evacuation scheduling. Experimental results on a real case study show that the two-stage approach produces better primal bounds than the MIP model and is two orders of magnitude faster; It also produces dual bounds stronger than the linear relaxation of the MIP model. Finally, simulations of the evacuation demonstrate that convergent evacuation plans outperform existing approaches for realistic driver behaviors.

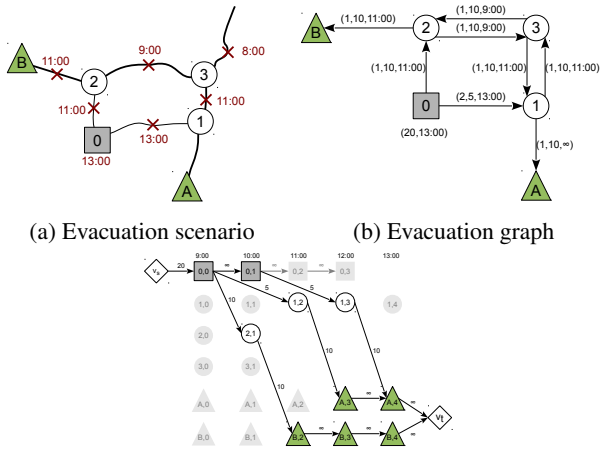
Introduction

Planning large-scale evacuations is central to the preparedness and response to many disasters including industrial hazards, floods, bushfires, hurricanes, and tsunamis. It ensures the safety of populations at threat by giving evacuees clear instructions on how to reach a set of safe zones. Evacuation planning is computationally challenging as it needs to account for the disaster, traffic network, residential zones and their population, and human behavior, among others. Given a threat or a set of threat scenarios, an evacuation plan must ideally produce two outputs: (1) a traffic management plan, including detailed evacuation routes and management points, and (2) an evacuation schedule, specifying when residential zones should evacuate. Most automated approaches to evacuation planning only address these factors partially and assume that individuals will decide their routes, destinations, and times for departure to reach a global optimum. The resulting plan essentially becomes a self-evacuation, where each individual makes their own decisions about whether and when to evacuate, using their preferred route and transportation mode. Self-evacuations increase the risk of congestion, since a large fraction of evacuees may choose the same evacuation routes or routes that cross or diverge at intersections.

This paper focuses on controlled evacuations in which authorities give evacuation orders to people under threat. Controlled evacuations ensure that each residential zone is assigned a specific route and time to evacuate, leading to the most effective utilization of the road network. These approaches significantly improve the quality of evacuation plans, evacuating more people over a given horizon or decreasing the overall evacuation time. However, specifying an evacuation route for each residential area is not sufficient to control an evacuation fully: Two evacuation routes can merge, possibly share a few road segments, and then diverge, leaving evacuees with a choice at the fork. Evidence collected during evacuations demonstrates that drivers hesitate when approaching a fork (Townsend 2006), leading to additional congestion.

To address this limitation, this paper introduces the concept of *convergent evacuation plans*, which assigns an evacuation route to each residential zone and ensures that all evacuation routes converge to the safe zones. Convergent evacuation plans can be fully controlled and remove a key source of congestion by ensuring that drivers never face a choice between two onwards evacuation routes. Optimal convergent plans can be obtained by a MIP model using a time-indexed flow formulation. Unfortunately, this model cannot tackle realistic-scale evacuations. To remedy this issue, the paper proposes a two-stage approach that separates the choice of evacuation routes from the scheduling of the evacuation. More precisely, the first stage is a tree design problem that selects a set of convergent routes and produces an upper bound on the number of people who can reach safety over a time horizon. The second stage is a flow scheduling problem that determines how to evacuate the residential zones optimally given the evacuation routes produced in the first stage. Experimental results on a real case study demonstrate the benefits of the two-stage approach. In particular:

- The tree design problem provides dual bounds that are



(a) Evacuation scenario (b) Evacuation graph

(c) Time-expanded graph

Figure 1: Modeling of an Evacuation Planning Problem.

stronger than the linear relaxation of the MIP model;

- The two-stage approach produces better primal bounds than the MIP model and is two orders of magnitude faster, producing high-quality convergent evacuation plans for 70,000 people in less than a minute;
- Convergent evacuation plans outperform traditional approaches even for minimal driver hesitations at forks.

Convergent Evacuation Planning

The input of the evacuation problem is an *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 the set of evacuation, transit, and safe nodes, and $\mathcal{A}$ is the set of arcs. Each evacuation node i is characterized by a number of evacuees d_i and an evacuation deadline $\bar{f}_i$, while each arc e is associated with a triple (s_e, u_e, f_e), where s_e is the travel time, u_e is the capacity, and f_e is the time at which the arc becomes unavailable. For modeling purposes, we assume that all evacuation (resp. safe) nodes are connected to a super source v_s (resp. sink v_t). Note that $u_{(v_s,i)} = d_i$.

Figure 1 illustrates these concepts. Figure 1a presents an evacuation scenario with one evacuation node (0) and two safe nodes (A and B). The evacuation node 0 must be evacuated by 13:00 and different arcs become unavailable at different times (for instance, $(2, 3)$ is cut at 9:00). Figure 1b specifies an evacuation graph based on this scenario. Here evacuation node 0 has a demand of 20 and a deadline of 13:00. The arc $(0, 1)$ has a travel time of 1, a capacity of 10, and becomes unavailable at 11:00.

To capture the spatio-temporal aspects of evacuation planning, this paper discretizes the planning horizon $\mathcal{H}$ into time steps of identical length $[0..h]$ and defines a *time-expanded graph* $\mathcal{G}^{\mathrm{x}} = (\mathcal{N}^{\mathrm{x}} = \mathcal{E}^{\mathrm{x}} \cup \mathcal{T}^{\mathrm{x}} \cup \mathcal{S}^{\mathrm{x}}, \mathcal{A}^{\mathrm{x}})$. $\mathcal{G}^{\mathrm{x}}$ is constructed by duplicating each node in $\mathcal{N}$ for each time step. $\mathcal{A}^{\mathrm{x}}$ contains an arc $e_t = (i_t, j_{t+s_{(i,j)}})$ for every time step t in which e is available to model the transfer of evacuees from node i at time t to node j at time $t + s_{(i,j)}$. In addition, arcs with infinite capacity are added to model evacuees waiting at evacuation and safe nodes. Figure 1c illustrates the time-expanded graph for the evacuation graph presented earlier. Note that some nodes may not be connected to either the source or the sink (in light grey in this example), and can therefore be removed to reduce the graph size.

The goal of convergent evacuation planning is to find a subgraph of the evacuation graph that maximizes the flow of evacuees and is fully controllable. A graph is *convergent* if, for every node $i \in \mathcal{N}$, the outdegree of i is 1. It is *connected* if, for every evacuation node $i \in \mathcal{E}$, there is a path from the source v_s to the sink v_t going through i. Observe that if we reverse the arcs direction and ignore v_s, these paths form a tree rooted at v_t with evacuation nodes as leaves. An evacuation graph that is both connected and convergent is fully controllable from an operational perspective.

Proposition 1. *Let $\mathcal{G}$ be a connected evacuation graph. There exists a connected and convergent subgraph $\mathcal{T}$ of $\mathcal{G}$.*

Proposition 2. *There is a unique path from an evacuation node to the sink in every connected and convergent evacuation graph.*

The Convergent Evacuation Planning Problem (CEPP) can now be formally stated.

Definition 1 (CEPP). *Given a connected evacuation graph $\mathcal{G}$ as input, the Convergent Evacuation Planning Problem (CEPP) consists of finding a convergent subgraph $\mathcal{T}$ of $\mathcal{G}$ whose associated time-expanded graph $\mathcal{T}^x$ maximizes the flow from v_s to v_t.*

Literature Review

Despite its practical importance, evacuation planning has received only limited interest from the research community. As defined by Hamacher and Tjandra (2002), evacuation planning can be tackled using either *microscopic* or *macroscopic* approaches. Microscopic approaches focus on modeling and simulating the individual behaviors, movements, and interactions of evacuees. Macroscopic approaches, such as the one presented in this study, aggregate evacuees and model their movements as a flow in the evacuation graph. Evacuation planning is related to dynamic network flow problems, also referred to as flows over time, where the time required to traverse an edge is modeled explicitly. Ford and Fulkerson (1958) introduced the Maximum Dynamic Network Flow Problem (MDFP), which maximizes the flow from a single source to a single sink within a specific time horizon. Their polynomial algorithm first computes the maximum flow over one time period. It then uses flow decomposition to determine the s-t paths on the network. Finally, it creates temporally repeated flows through the static network until the demand is satisfied or the time horizon is reached. Burkard, Dlaska, and Klinz (1993) introduced the quickest flow problem (QFP) which aims at finding the shortest time horizon to transmit a certain amount of flow from a source to a sink. The authors demonstrate that it can be reduced to the MDFP using binary search to reduce the horizon as much as possible. Hoppe and Tardos (2000) generalized the QFP to several sources and sinks, defining the quickest transshipment problem (QTP), and presented a

polynomial algorithm for the QTP by taking an approach inspired from temporally repeated flows. A Maximum Flow Tree (MFT) is a spanning tree T in a network G such that every rooted path in T is at maximum capacity. The seminal work by Gouda and Schneider (1995) presents a method to compute a MFT that builds a spanning tree with maximum capacity paths, and updates the tree when a connection between two nodes is established. Siachalou and Georgiadis (2005) propose algorithms to compute solutions to the constrained widest multicast tree problem, which extends the MFTP with additional constraints on the tree.

However, in the context of evacuation, MDFPs and MFTs do not produce actionable plans that associate a single path to a safe zone with each evacuation node. In fact, few studies attempt to design actionable evacuation plans. Huibregtse et al. (2011) proposed a two-stage algorithm that first generates a set of evacuation routes and feasible departure times, and then assigns a route and time to each evacuated area using an ant colony optimization algorithm. In subsequent work, the authors studied the robustness of the produced solution (Huibregtse, Bliemer, and Hoogendoorn 2010), and strategies to improve the compliance of evacuees (Huibregtse, Hegyi, and Hoogendoorn 2012). Even, Pillac, and Van Hentenryck (2014) and Pillac, Van Hentenryck, and Even (2014) proposed a scalable conflict-based path-generation algorithm that produces actionable evacuation plans. These plans jointly schedule the evacuation and the selection of contraflow roads.

This paper goes one step further and proposes algorithms for convergent evacuation paths which are fully controllable. The algorithms combine the design of a set of *evacuation routes* and an *evacuation schedule* ensuring the *convergence of the flow evacuees*.

The MIP Model

This section presents a MIP model for solving the CEPP. The model uses a binary variable x_e to denote whether arc $e \in \mathcal{G}$ is selected and a continuous variable φ_{e_t} for the flow on arc $e_t \in \mathcal{G}^{\text{x}}$. The MIP model is formulated as follows:

$$\max \sum_{e_t \in \delta^+(v_s)} \varphi_{e_t} \tag{1}$$

s.t.

$$\sum_{e_t \in \delta^-(i)} \varphi_{e_t} - \sum_{e_t \in \delta^+(i)} \varphi_{e_t} = 0 \qquad \forall i \in \mathcal{N}^{\text{x}} \tag{2}$$

$$\sum_{e \in \delta^+(i)} x_e \leq 1 \qquad \forall i \in \mathcal{N} \tag{3}$$

$$\varphi_{e_t} \leq x_e . u_{e_t} \qquad \forall e \in \mathcal{A}, \forall t \in \mathcal{H} \tag{4}$$

$$\varphi_{e_t} \geq 0 \qquad \forall e_t \in \mathcal{A}^{\text{x}} \tag{5}$$

$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{A} \tag{6}$$

where $\delta^-(i)$ (resp. $\delta^+(i)$) denotes the set of incoming (resp. outgoing) edges of node i. Constraints (2) ensure flow conservation through the network, constraints (3) enforce convergence of the evacuation paths, constraints (4) link the arc and flow variables, while the objective (1) maximizes the total evacuee flow.

The Proposed Approach

Computational experiments show that the MIP formulation quickly becomes intractable for real-sized instances. This section presents a two-stage approach that separates the choice of the converging paths from the flow scheduling.

Tree Design Problem

The first stage of the proposed approach is a Tree Design Problem (TDP). Intuitively, the TDP is an abstraction of the CEPP where the evacuation flow is aggregated over time, avoiding the need to reason about the time-expanded evacuation graph. Let $\mathcal{G}^+$ be the evacuation graph $\mathcal{G}$ where the arc capacities have been summed over the time horizon. The TDP consists of finding a convergent subgraph of $\mathcal{G}^+$ that maximizes the flow from the source to the sink.

The TDP can be formulated as a MIP model with a binary variable y_e and a continuous flow variable ψ_e for each arc $e \in \mathcal{A}$. The binary variable denotes whether arc e is selected, while the continuous variable represents the aggregated flow on arc e. The TDP can be formulated as follows:

$$\max \sum_{e \in \delta^+(v_s)} \psi_e \tag{7}$$

s.t.

$$\sum_{e \in \delta^-(i)} \psi_e - \sum_{e \in \delta^+(i)} \psi_e = 0 \qquad \forall i \in \mathcal{N} \tag{8}$$

$$\sum_{e \in \delta^+(i)} y_e \leq 1 \qquad \forall i \in \mathcal{N} \tag{9}$$

$$\psi_e \leq y_e \sum_{t \in \mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A} \tag{10}$$

$$\psi_e \geq 0 \qquad \forall e \in \mathcal{A} \tag{11}$$

$$y_e \in \{0,1\} \qquad \forall e \in \mathcal{A} \tag{12}$$

Constraints (8) and (9) ensure flow conservation and a convergent plan, constraints (10) impose the flow capacity on each arc, and the objective (7) maximizes the flow.

Theorem 1. *The optimal solution of TDP is an upper bound to the CEPP.*

Proof. We show that an optimal solution to the CEPP can be transformed into a solution of the TDP with the same objective value. Let $\Phi = \{\varphi_{e_t}, x_e\}_{e_t \in \mathcal{A}^{\text{x}}, e \in \mathcal{A}}$ be an optimal solution to the CEPP and $z(\Phi)$ be its objective value. Consider the following candidate solution to the TDP: $\Psi = \{\psi_e = \sum_{t \in \mathcal{H}} \varphi_{e_t}, y_e = x_e\}_{e \in \mathcal{A}}$. By definition of the flows, the objective value of Ψ is equal to $z(\Phi)$. We now show that Ψ satisfies all constraints of the TDP. Since Φ is a solution

to the CEPP, it satisfies constraints (2) and we have

$$\sum_{e_t \in \delta^-(i)} \varphi_{e_t} - \sum_{e_t \in \delta^+(i)} \varphi_{e_t} = 0 \qquad \forall i_t \in \mathcal{N}^{\mathrm{x}}$$

$$\Rightarrow \sum_{e \in \delta^-(i)} \sum_{t \in \mathcal{H}} \varphi_{e_t} - \sum_{e \in \delta^+(i)} \sum_{t \in \mathcal{H}} \varphi_{e_t} = 0 \qquad \forall i \in \mathcal{N}$$

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

and Ψ satisfies constraints (8). Since Φ satisfies constraints (3), it follows that Ψ satisfies constraints (9). Finally, since Φ satisfies constraints (3), we have

$$\varphi_{e_t} \leq x_e . u_{e_t} \qquad \forall e \in \mathcal{A}, \forall t \in \mathcal{H}$$

$$\Rightarrow \sum_{t \in \mathcal{H}} \varphi_{e_t} \leq x_e \sum_{t \in \mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A}$$

$$\Rightarrow \psi_e \leq y_e \sum_{t \in \mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A}$$

and Ψ satisfies (10). The result follows. □

Flow Scheduling

An optimal solution to the TDP defines a connected and convergent evacuation graph $\mathcal{T}$ used in the second stage to schedule the flow of evacuees. The second stage is a Flow Scheduling Problem (FSP) that maximizes the flow in the time-expanded graph $\mathcal{T}^{\mathrm{x}}$ associated with $\mathcal{T}$.

The FSP avoids the creation of the time-expanded graph by reasoning about paths from the evacuation nodes to the sink. By Propositions (1) and (2), the TDP produces a convergent and connected subgraph which includes a unique path from each evacuation node to the sink. We denote by Ω this set of paths and use the following notations: τ_p^e is the time taken to reach edge e from the source node using path $p \in \Omega$, $\mathcal{H}_p = \left[0..h - \tau_p^{v_t}\right]$ is the set of departure times that would allow evacuees to reach safety by using path p, p_i is the path associated with evacuation node $i \in \mathcal{E}$ in the TDP solution, and $w(e)$ is the set of paths using arc e.

The FSP uses a continuous variable χ_t^p for each path $p \in \Omega$ and $t \in \mathcal{H}_p$ to represent the flow leaving at time t along path p. The FSP can then be specified by the following linear program:

$$\max \quad \sum_{p \in \Omega} \sum_{t \in \mathcal{H}_p} \chi_p^t \qquad (13)$$

$$\text{s.t.} \sum_{t \in \mathcal{H}_p} \chi_{p_i}^t \leq d_i \qquad \forall i \in \mathcal{E} \qquad (14)$$

$$\sum_{\substack{p \in \omega(e) \\ t - \tau_p^e \in \mathcal{H}_p}} \chi_p^{t - \tau_p^e} \leq u_{e_t} \qquad \forall e \in \mathcal{A}, t \in \mathcal{H} \qquad (15)$$

$$\chi_p^t \geq 0 \qquad \forall p \in \Omega, t \in \mathcal{H}_p \qquad (16)$$

Constraints (14) ensure that the number of evacuees along path p_k does not exceed the number of evacuees d_k in evacuation node k, while constraints (15) enforce arc capacity by reasoning about the number of evacuees that can reach arc e at time t. The objective (13) maximizes the total flow along all paths and all times in the horizon.

In the following, we denote by $\texttt{TDP}(\mathcal{G}, \mathcal{H})$ an optimal solution to the stage 1 (TDP) given an input evacuation graph $\mathcal{G}$ and horizon $\mathcal{H}$ and by $\texttt{FSP}(\Psi, \mathcal{H})$ an optimal solution to the stage two (FSP) given a solution Ψ to the TDP and horizon $\mathcal{H}$, and $\texttt{CEPP}(\mathcal{G}, \mathcal{H})$ an optimal solution to the CEPP MIP. We also use $z(\sigma)$ to denote the objective value of a solution σ to the CEPP, TDP, or FSP.

Corollary 1. *Let $\mathcal{G}$ be a connected evacuation graph, $\mathcal{H}$ a time horizon. We have $z(\texttt{TDP}(\mathcal{G}, \mathcal{H})) \geq z(\texttt{CEPP}(\mathcal{G}, \mathcal{H})) \geq z(\texttt{FSP}(\texttt{TDP}(\mathcal{G}, \mathcal{H}), \mathcal{H}))$.*

Algorithm 1 The two-stage Algorithm TDFS for the CEPP

Require: $\mathcal{G}$ the evacuation graph, $\mathcal{H}$ the time horizon
Ensure: A convergent evacuation plan
1: $t^* \leftarrow \min\{t \in \mathcal{H} \mid z(\texttt{TDP}(\mathcal{G}, [0..t])) = z(\texttt{TDP}(\mathcal{G}, \mathcal{H}))\}$
2: **return** $\texttt{FSP}(\texttt{TDP}(\mathcal{G}, [0..t^*]), \mathcal{H})$

The two-stage Approach

When the time horizon is too large, many convergent evacuation plans can be optimal solutions to the TDP. However, these plans often differ substantially in their ability to deal with congestion. In practice, using optimal solutions to the TDP with a tighter time horizon produces significantly better plans for the FSP. As a result, the two-stage approach TDFS computes the tightest time horizon t^* that preserves the optimal solution to the TDP and solves the FSP using the TDP solution obtained for t^*.

Algorithm 1 implements this idea and formalizes our two-stage approach. Observe that the FSP receives the full horizon, since the TDP is an approximation to the CEPP and it may not be possible to schedule the TDP flow in the time-expanded graph for t^*. The tightest time horizon can be obtained by dichotomic search over the horizon using the TDP as a subproblem.

Minimizing Clearance Time

In evacuation planning, it is often desirable to determine the earliest time by which all evacuees can reach safety. More precisely, the goal is to compute the minimum clearance time h^* defined as

$$h^* = \min \left\{ h \in \mathbb{R} \mid z(\texttt{CEPP}(\mathcal{G}, [0..h])) = \sum_{i \in \mathcal{E}} d_i \right\}.$$

To that purpose, we adapt the TDP model (7-12) by introducing a new continuous variable h representing the length of the horizon, setting the objective to $\min h$, and by substituting (10) with (10b) and adding (17):

$$\psi_e \leq h \max_{t \in \mathcal{H}} u_{e_t} \qquad \forall e \in \mathcal{A} \qquad (10b)$$

$$\psi_{(v_s, i)} = d_i \qquad \forall i \in \mathcal{E} \qquad (17)$$

The resulting model, namely TDP-CT, provides a convergent subgraph and a lower bound on the clearance time.

Algorithm 2 presents the TDFS-CT heuristic which finds a convergent evacuation plan of minimum clearance time

Algorithm 2 The two-stage Algorithm TDFS-CT for the CEPP with Minimization of the Clearance Time

Require: $\mathcal{G}$ the evacuation graph
Ensure: A convergent evacuation plan of minimum clearance time
1: $\Psi \leftarrow \texttt{TDP–CT}(\mathcal{G})$
2: $h^* \leftarrow \min\left\{t \in \mathbb{R} \mid z(\texttt{FSP}(\Psi, [0..t])) = \sum_{i \in \mathcal{E}} d_i\right\}$
3: **return** $\texttt{FSP}(\Psi, [0..h^*])$

given a solution of TDP-CT. In our implementation, step 2 uses a dichotomic search initialized with the lower bound provided by TDP-CT.

Experimental Results

This section presents experimental results for the the evacuation of the Hawkesbury-Nepean (HN) floodplain, located North-West of Sydney. We consider a 1-in-200 years flood that would require the evacuation of 70,000 people. The HN evacuation graph contains 80 evacuated nodes, 5 safe nodes, 184 transit nodes, and 580 edges. The time horizon $\mathcal{H}$ spans 10 hours discretized into 5 minutes time-steps. The experimental results also consider a class of instances HN80-Ix using the HN evacuation graph but a number of evacuees scaled by a factor x $\in [1.1, 3.0]$ to reflect projected population growth over the next decades. The algorithms were implemented using JAVA 7 and GUROBI 5.6 and the results were obtained on 64bits machines with 2.8GHz AMD 6-Core Opteron 4184 and 32Gb of RAM.

Dual Bounds

Table 1 compares the bounds provided by the TDP and the Linear Relaxation (LR) of the MIP model for the HN80 instances. Both models provide an upper bound on the percentage of evacuated vehicles. In this table, columns Perc. Evac. report the percentage of evacuated residents for both models, columns CPU report the execution time in seconds (s), while the column Gap TDP reports the gap relative to the TDP solution $\frac{z(\texttt{TDP}(\mathcal{G},\mathcal{H}))-z(\star)}{z(\star)}$ where $z(\star)$ is the total flow found by model $\star$ (here $\star$ is LR).

The results indicate that the TDP gives tighter dual bounds. On average, the TDP produces a 3.5% improvement in dual bounds, while improving LR by more than 10% on the larger instances. The TDP is also 9 times faster than LR on average.

Primal Bounds

Table 2 compares the two-stage TDFS with the best solution found by the MIP model in 24 hours. For each iteration of TDFS, we allocate 30s for each call to TDP. The results indicate that, on average, TDFS evacuates 1.3% more people than CEPP in about 14s. The average optimality gap is 0.2% and the worst-case optimality gap is 0.7%. In contrast, even with a 24 hours time limit, the MIP model has a worst-case optimality gap of 7.9% and an average gap of 1.7%.
Figure 2 shows the evacuation percentage for the MIP model (solid grey columns) and TDFS (solid white column), as well as the MIP gap (hashed) for all HN80-Ix instances under various time limits for te MIP model. The light, medium, and dark grey colors represent the MIP incumbent solution and gap after 1, 4, and 24 hours. The gap between the MIP objective value and the best bound highlights the slow convergence of the MIP solver.

Instance	TDP CPU (s)	TDP Perc. Evac.	LR CPU (s)	LR Perc. Evac.	LR Gap TDP
HN80	0.2	100%	7.1	100%	0.0%
HN80-I1.1	0.3	100%	7.9	100%	0.0%
HN80-I1.2	0.2	100%	11.2	100%	0.0%
HN80-I1.4	0.3	100%	9.8	100%	0.0%
HN80-I1.7	0.7	100%	13.1	100%	0.0%
HN80-I2.0	2.6	96.2%	12.4	100%	-3.8%
HN80-I2.5	5.0	81.1%	13.2	91.2%	-11.1%
HN80-I3.0	0.6	68.1%	13.7	76.1%	-10.5%
Average	*1.2*	*93.2%*	*11.0*	*95.9%*	*-3.2%*

Table 1: Dual Bounds for the HN80-I Instances.

Instance	TDFS CPU (s)	TDFS Perc. Evac.	TDFS Gap TDP	MIP 24h Perc. Evac.	MIP 24h Gap TDP
HN80	33.4	100%	0.0%	100%	0.0%
HN80-I1.1	1.3	100%	0.0%	100%	0.0%
HN80-I1.2	1.0	100%	0.0%	100%	0.0%
HN80-I1.4	6.4	100%	0.0%	100%	0.0%
HN80-I1.7	47.6	100%	0.0%	99.1%	0.9%
HN80-I2.0	4.1	95.5%	0.7%	89.1%	7.9%
HN80-I2.5	6.6	80.7%	0.5%	77.5%	4.7%
HN80-I3.0	1.5	67.7%	0.6%	68.0%	0.2%
Average	*14.1*	*93.0%*	*0.2%*	*91.7%*	*1.7%*

Table 2: Primal Bounds for the HN80-I Instances.

Clearance Time Minimization

Table 3 presents results obtained with the TDFS-CT approach on the HN80 instances. The second column reports the optimal clearance time obtained using a dichotomic search using a free-flow model (FF-CT), which does not restrict the flow of evacuees (Pillac, Van Hentenryck, and Even 2013). The third and fourth columns report results for the TDP-CT model run independently, and the fifth to seventh column present the results from the TDFS-CT approach, including the absolute optimality gap with respect to the lower bound provided by TDP-CT. In these results, the TDP-CT model is stopped after 30s when the duality gap is lower than 1%.

The results show the competitiveness of the TDFS-CT approach which finds a clearance time close to the lower bound provided by the TDP-CT. The quality of the solutions is particularly high, given that the TDP-CT model ignores travel times along the arcs and that it takes between 10 and 55 min to reach a safe node from any evacuated area. The results also demonstrate the inadequacies of free-flow approaches, which greatly underestimate the time required to evacuate all vehicles in an orderly fashion. Note that the average running time of TDFS-CT is about 110s, of which an average

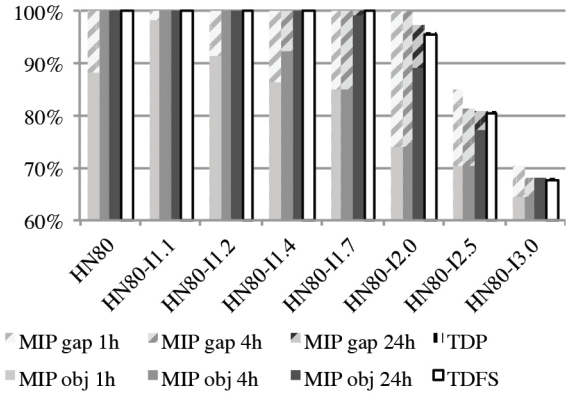


Figure 2: Primal Bounds as a Function of Solving Time.

Instance	FF-CT	TDP-CT		TDFS-CT		
	CT (min)	CPU (s)	CT (min)	CPU (s)	CT (min)	Gap (min)
HN80-I1.0	260	105	300	115	345	48
HN80-I1.1	280	30	330	39	375	48
HN80-I1.2	300	108	360	122	410	53
HN80-I1.4	340	30	420	47	470	54
HN80-I1.7	405	220	510	233	575	69
HN80-I2.0	465	35	595	63	645	50
HN80-I2.5	570	63	745	92	820	76
HN80-I3.0	670	120	895	171	935	43

Table 3: Clearance Time on the HN80-I Instances.

of about 90s is spent solving the TDP-CT model, and an average of about 20s is devoted to performing the dichotomic search to find the minimum clearance time.

Benefits of Convergent Evacuation Plans

Figure 3 illustrates the benefits of convergent evacuation plans. The chart compares the clearance times of TDFS-CT with an instrumented mesoscopic simulation of the evacuation plans produced by the CPG algorithm (Pillac, Van Hentenryck, and Even 2014). CPG assigns a single evacuation path to every evacuation node but allows for divergent paths. The simulation instrumentation adds a delay at each fork for each vehicle, capturing the driver hesitation. The chart shows the effect of this delay on clearance times. The results show that TDFS-CT starts outperforming CPG for an average delay as small as 0.75s. The benefits of TDFS becomes substantial even for delays as small as 2s.

Conclusion

This paper introduced the concept of convergent evacuation plans to produce fully controllable evacuations avoiding forks. Forks lead to congestions in practice as drivers slow down to consider the alternatives ahead of them. Additionally, the presence of forks makes it harder to guarantee that evacuees will actually follow the evacuation plan.

The paper formalized the Convergent Evacuation Planning Problem (CEPP) and presented a MIP model for finding a convergent evacuation plan maximizing the number of people evacuated. To remedy its scalability issues, the paper proposed a two-stage approach, separating the design of the convergent evacuation routes from the scheduling of evacuees along these routes. The first stage is a Tree Design Problem (TDP) which aggregates arc capacities over time and avoid discretizing time. The second stage is a flow scheduling problem (FDP) that chooses when to evacuate a residential area along a path. Optimal solutions to the TDP are upper bounds to the optimal solutions to the CEPP. The paper also presented results on minimizing the clearance times using a variant of the two-stage approach.

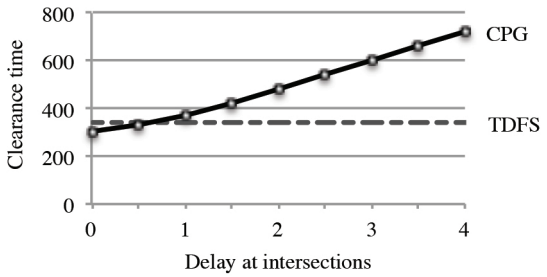


Figure 3: The Effect of Fork Delays on Clearance Times (HN80).

Experimental results on a real case study validate the benefits of the approach. In particular, they show that: (1) The TDP provides stronger dual bounds than the linear relaxation of the MIP model; (2) The two-stage approach provides high-quality solutions with average and worst-case optimality gaps of 0.2% and 0.7% in less than a minute of CPU Time. In contrast, the MIP model provides solutions with average and worst-case optimality gaps of 1.7% and 7.9% even with a time limit set to 24 hours; (3) The benefits of the two-stage approach materialize as soon as a delay of 0.75s is introduced at a fork when simulating the results of state-of-the-art evacuation planning tools on a mesoscopic simulator. The benefits become subtantial when the delay increases: The clearance time doubles when the delay is about 4s.

Future work will focus on extending the models to integrate contraflow decisions and lane separation, build accurate behavioral models of evacuees when responding to evacuation orders, and exploit these models inside the evacuation planning algorithms. Contraflows and lane separations would give more flexibility to the optimisation algorithm to exploit routes that would not be convergent otherwise. Finally, observe that our algorithms can be generalized to allow forks where the resulting increased capacity is beneficial regardless of the slowdown produced.

Acknowledgments

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

Burkard, R.; Dlaska, K.; and Klinz, B. 1993. The quickest flow problem. *Methods and Models of Operations Research* 37:31–58.

Even, C.; Pillac, V.; and Van Hentenryck, P. 2014. Nicta evacuation planner: Actionable evacuation plans with contraflows. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2014)*.

Ford, L. R., and Fulkerson, D. R. 1958. Constructing maximal dynamic flows from static flows. *Operations Research* 6(3):419–433.

Gouda, M. G., and Schneider, M. 1995. Maximum flow routing. Technical report, Department of Computer Science, University of Nevada, Las Vegas.

Hamacher, H. W., and Tjandra, S. A. 2002. Mathematical modelling of evacuation problems: A state of art. In Schreckenberger, M., and Sharma, S., eds., *Pedestrian and Evacuation Dynamics*. Springer Verlag. 227–266.

Hoppe, B., and Tardos, E. 2000. The quickest transshipment problem. *Mathematics of Operations Research* 25(1):36–62.

Huibregtse, O. L.; Hoogendoorn, S. P.; Hegyi, A.; and Bliemer, M. C. J. 2011. A method to optimize evacuation instructions. *OR Spectrum* 33(3):595–627.

Huibregtse, O. L.; Bliemer, M. C.; and Hoogendoorn, S. P. 2010. Analysis of near-optimal evacuation instructions. *Procedia Engineering* 3:189–203.

Huibregtse, O.; Hegyi, A.; and Hoogendoorn, S. 2012. Blocking roads to increase the evacuation efficiency. *Journal of Advanced Transportation* 46(3):282–289.

Pillac, V.; Van Hentenryck, P.; and Even, C. 2013. A conflict-based path-generation heuristic for evacuation planning. Technical Report VRL-7393, NICTA, Melbourne.

Pillac, V.; Van Hentenryck, P.; and Even, C. 2014. A path-generation matheuristic for large scale evacuation planning. In Blesa, M.; Blum, C.; and Voss, S., eds., *Hybrid Metaheuristics*, volume 8457 of *Lecture Notes in Computer Science*, 71–84. Springer.

Siachalou, S., and Georgiadis, L. 2005. Algorithms for precomputing constrained algorithms for precomputing constrained widest paths and multicast trees. *IEEE/ACM Transactions on Networking* 13(5):1174–1187.

Townsend, F. F. 2006. The federal response to hurricane Katrina - lessons learned. Technical report, The White House, Washington.