Optimal design of rapid evacuation strategies in constrained urban transport networks

Corresponding Author:
Jose Javier Escribano-Macias
PhD Student in Sustainable Civil Engineering
Department of Civil and Environmental Engineering
Imperial College London
SW7 2BU, UK
jose.escribano-macias11@imperial.ac.uk

Panagiotis Angeloudis
Senior Lecturer in Transport Systems and Logistics
Centre for Transport Studies
Department of Civil and Environmental Engineering
Imperial College London
SW7 2BU, UK
p.angeloudis@imperial.ac.uk

Ke Han
Lecturer in Transport Operations and Logistics
Centre for Transport Studies
Department of Civil and Environmental Engineering
Imperial College London
SW7 2BU, UK
k.han@imperial.ac.uk
Optimal design of rapid evacuation strategies in constrained urban transportation networks

ABSTRACT

Evacuations constitute essential life-saving processes during any disaster response activity, aiming to safeguard the well-being of populations in peril. Their effectiveness depends on the structure of the coordination mechanisms that may be in place, evacuee numbers, and the state of the underlying transportation network. Network capacity constraints and the inevitable surge in demand during evacuations often lead to congestion, which is regarded as unavoidable and severely delays the process. This paper presents a hybrid simulation-optimisation methodology to optimise evacuation response strategies through demand staging. We introduce a pre-planning model that evaluates policies, using a low-level dynamic traffic assignment model that captures the effects of congestion, queuing and vehicle spillback. Optimal evacuation strategies are determined using derivative-free optimisation algorithms, applied to an evacuation problem based on a benchmark dataset. The effects of varying the number of activated paths and the frequency of departure are observed. Our analysis indicates that departure time scheduling is a promising method to improve evacuation efficiency when compared to a worst-case benchmark scenario.

Keywords: simulation-optimisation, disaster evacuation, genetic algorithms, dynamic traffic assignment

1. INTRODUCTION

Evacuations are disaster response mechanisms aiming to temporarily relocate populations, before, during and after large-scale disasters while seeking to minimise potential loss of life. Given the inherent managerial, societal, financial, behavioural, and operational aspects of evacuations, adequate planning is required to ensure their effectiveness, and that individual safety is ensured. Even though it is difficult to predict the number of lives that are likely to be saved through a specific evacuation strategy, there is evidence that countries with extensive disaster preparedness response have been able to reduce casualties from natural disasters to nearly zero (Hallegatte 2012).

Key among the factors impeding evacuation processes is the unique traffic distribution caused by the sudden surge in traffic with patterns that contravene normal traveller behaviours. Even with extensive warning, the mass mobilisation imposes strains on the transport network, particularly considering the limited time, resources available and increasing urban population (Thomas & Kopczak 2007). As a result, network disruption is among the greatest challenges encountered during emergency evacuations and is aggravated by spillbacks, shockwaves and gridlocks. This phenomenon was illustrated during the Tohoku earthquake and subsequent tsunami in 2011, where according to Yun & Hamada (2015), 51% of casualties were already evacuating, of which it is estimated that 30% had problems reaching shelters due to network congestion. An overview of recent studies on high profile natural disasters, their impacts and problems encountered during evacuation is provided in Table 1.

As is the case with ordinary traffic, many evacuation strategies rely on capacity expansion measures to preclude the occurrence of congestion effects. Among them, contraflow and crossing elimination have been previously found to result in capacity increases of up to 70% and 40%, respectively (Wolshon 2001; Cova & Johnson 2003). As the optimal design of such configurations is an NP-complete problem, heuristics have been previously used to minimise computational times (Tuydes & Ziliaskopoulos 2006; Shekhar & Min 2008). Such techniques, however, rely on prior familiarity with evacuation plans and are likely to increase safety risks in scenarios where evacuees are unaware or non-compliant (Xie et al. 2010).
Table 1
List of recent natural disasters and evacuation problems.

<table>
<thead>
<tr>
<th>Disaster</th>
<th>Area</th>
<th>Date</th>
<th>Deaths</th>
<th>Evacuation</th>
<th>Problems</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>PH</td>
<td>2013</td>
<td>6,300</td>
<td>800,000</td>
<td>Damage to evacuation sites</td>
<td>Mahar et al. (2015)</td>
</tr>
<tr>
<td>F</td>
<td>IN</td>
<td>2013</td>
<td>4,000</td>
<td>24,000</td>
<td>Evacuation order not issued.</td>
<td>Rautela (2013)</td>
</tr>
<tr>
<td>EQ</td>
<td>ID</td>
<td>2012</td>
<td>10</td>
<td>3 million</td>
<td>Traffic congestion</td>
<td>Munawir (2012)</td>
</tr>
<tr>
<td>EQ-TS</td>
<td>JP</td>
<td>2011</td>
<td>16,000</td>
<td>400,000</td>
<td>Damage to transport infrastructure</td>
<td>Hasegawa (2013)</td>
</tr>
<tr>
<td>EQ</td>
<td>HT</td>
<td>2010</td>
<td>230,000</td>
<td>482,000</td>
<td>Lack of warning, damage to transport</td>
<td>Margesson &amp; Taft-Morales (2010)</td>
</tr>
<tr>
<td>H</td>
<td>US</td>
<td>2005</td>
<td>1,800</td>
<td>400,000</td>
<td>Over 100,000 to 300,000 did not or could not evacuate.</td>
<td>Wolshon (2006)</td>
</tr>
<tr>
<td>EQ-TS</td>
<td>SA</td>
<td>2004</td>
<td>220,000</td>
<td>N/A</td>
<td>Lack of warning, damage to transport</td>
<td>Fehr et al. (2005)</td>
</tr>
</tbody>
</table>

Table Legend

As an alternative, demand management has been explored as a means to limit congestion effects by restructuring the distribution of evacuees across the network (Abdelgawad & Abdulhai 2009). This is achieved by providing alternate safe sites to evacuees or by staging departures at their source (Chen & Zhan 2008).

The estimation of expected network congestion levels during an evacuation is not a trivial task. The occurrence and scale of gridlocks, spillbacks and shockwaves are dependent upon time and user behaviour, and as such cannot be integrated directly into linear optimisation algorithms without affecting model accuracy or computational performance (Lu et al. 2005). As a result, many previous studies on congestion prediction have adopted bi-level optimisation models that involve Stackelberg games. In such model, the upper-layer usually focuses on decisions taken by a “super-user”, who takes the role of the decision-maker and possesses perfect knowledge of the evacuation process. In practice, this role is assumed by emergency management and law enforcement organisations, with support from transport agencies (Urbina & Wolshon 2003). Conversely, the lower level model focuses on the effects of these decisions (Abdelgawad & Abdulhai 2009). Common implementations are usually based on traffic assignment models, which may vary in scale and detail (Table 2).

Table 2
Traffic assignment model types

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microscopic</td>
<td>Considers individual behaviours and congestion effects. Computationally expensive.</td>
<td>OREMS (Li et al. 2006)</td>
</tr>
<tr>
<td>Macroscopic</td>
<td>Vehicle movement modelled as fluid flows. Less emphasis on individual interactions.</td>
<td>DYNEV (Goldblatt 2004) and models by Friesz et al. (1993) and Han et al. (2013)</td>
</tr>
</tbody>
</table>
In the context of demand management, traffic models can develop system-optimal (SO), user equilibrium (UE) or shortest path (SP) assignments. Hobeika & Kim (1998) compared the effects of modelling UE and SP assignment in nuclear power plant evacuations using the MASSVAC emulation software. As is also the case in many other studies, SO assignment was not considered therefore limiting the effectiveness the resulting strategies. Sbayti & Mahmassani (2006) developed a method in which the upper-level feeds all evacuees’ origin-destination (OD) choices and initial time-dependent historic assignment paths to the lower level. The latter was based on a mesoscopic cell transmission model (CTM) featuring dynamic SO traffic assignment. Users were assumed to always follow the paths determined by the model, which is not usually the case in practice, and especially during evacuations.

Liu et al. (2006) propose the discretisation of the evacuation area into staged departure zones, instead of the node-based departure staging strategies that were previously used. Their analysis indicates that optimal results are obtained using a combination of urgency and anticipated demand patterns, rather than urgency alone. A similar problem is encountered by Bish et al. (2013), who formulate a mixed integer linear problem seeking to reduce network clearance times. This was based on a CTM and illustrated the positive effects of evacuation staging when sufficient advance warning is provided. Even though their model assumes that all users comply with the evacuation order, a key advantage over previous work is the adoption of household-level fidelity levels.

Mitchell & Radwan (2006) use a microscopic traffic assignment method with pre-trip and en-route choice models to optimise network clearance time. Chien & Korikanthimath (2007) presented a model that consider the total duration and any congestion delays of multistage evacuations. Results suggest that multistaging is an efficient operation to reduce evacuation time provided that the OD choices are defined by the “super-user” and that the disaster allows for flexibility in time. Chen & Zhan (2008) used agent-based simulation to evaluate the merits of a proposed evacuation staging strategy, which was found to perform well for geometric transportation networks in high-density areas. Their analysis indicates that the effectiveness of a specific evacuation strategy is heavily dependent upon road network structures and that staging is beneficial outside free-flow conditions. Finally, Abdelgawad & Abdulhai (2010) propose a spatiotemporal evacuation framework that minimises travel and wait times for evacuees.

The assumption that all individuals will comply with evacuation plans that are prepared by authorities is common across the literature. Another limitation of existing techniques is the reliance on approximate models that limit the accuracy and computational effectiveness of the lower-level models.

To address these gaps, we propose an assignment-optimisation framework composed of two elements that contribute to the development of an evacuation strategy, namely an evacuation decision model and a dynamic traffic model. The evacuation decision model solely controls the departure rates of evacuees a specified time horizon, while the dynamic traffic model evaluates network properties and traffic behaviour. We use an efficient numerical traffic model based on continuous time-dependent kinematic wave simulation, with consideration of congestion effects and the impact of user information on road traffic.

The two models are integrated using a Stackelberg game acting as a simulation-optimisation framework (Nguyen et al. 2014). Both consist of a two-step cyclical process, where the simulation model imitates real-world characteristics and the optimisation strategy determines the inputs that yield the best outputs. The main advantage of the framework is the capability of optimising a system when derivative information is not available (Amaran et al. 2016).

The mathematical formulation used to determine optimal evacuation strategies is provided in Chapter 2, followed by the traffic and evacuation routing models that are discussed in Chapter 3. The implementation of the overall algorithm is described in Chapter 4, alongside any modifications made to the underlying models to facilitate closer integration. A numerical example is also presented accompanied by an analysis with discussions regarding its performance and scalability. Conclusions and recommendations for future work are provided in Chapter 5.
2. Evacuation Strategy Model

The upper-level model in our framework is used to capture the decisions involved in the design of the evacuation strategy, with the objective to maximise evacuee numbers within a predefined timespan. Given the risk of non-compliance that was highlighted by Xie et al. (2010), we assume that the evacuation strategy adopts a departure staging mechanism, with users not prescribed a specific route but instead merely instructed when to vacate their premises. Once users are allowed to depart, they are free to travel towards any of the evacuation shelters (implemented as sink nodes) that have been made available. The following assumptions are adopted:

Assumption 1: We assume that evacuees use their road vehicle as a mode of transport.
We adopt this assumption in order to convert the model into a single-commodity assignment problem. We acknowledge that public transport can aid evacuation performance, especially among specific groups such as limited mobility evacuees (Urbina & Wolshon 2003) and that walking would be a likely mode choice in short distance evacuation (Wood & Schmidtlein 2012).

Assumption 2: The network is assumed to be initially empty.
This assumption follows from the deterministic nature of the model and its intended use as a strategic design tool. However, we recognise that background traffic is likely to affect the performance of the evacuation strategy alongside the availability of information system and the nature of hazard type (Zheng et al. 2010). In practice, we expect that evacuation planners would apply the framework in a range of problem instances, each pertaining to a different network state.

Assumption 3: Effects of shadow evacuation and intermediate trips are ignored.
Shadow evacuation considers self-evacuating users, accounting for up to 20% of evacuees for uncommon hazards, such as chemical or nuclear threats (Zeigler et al. 1981; Mitchell et al. 2007). We consider a common hazard occurrence for which users are familiar with existing evacuation procedures.

Assumption 4: Intermediate trips are not considered.
Intermediate trips made to an alternate destination before evacuating increase network clearance time, and are more common under smaller warning time (Murray-Tuite & Wolshon 2013; Liu & Murray-tuite 2011). An illustrative example is provided in the 2012 Indonesia Earthquake, where the evacuation was undermined by families collecting their children from the education centres (Munawir 2012; News24 2012). We assume that extensive warning time is given, minimising the impact of this phenomenon.

Assumption 5: The super-user is assumed to have perfect information of the traffic network.
The assumption is based on the existence of Intelligent Transport Systems that can update information on the transport network during the evacuation. Therefore, authorities can use traffic information feeds to collect data on the network state in real-time, providing evacuees with instructions that reflect that current state of the network.

Assumption 6: Users cannot change their evacuation route after departure.
During evacuation, evacuees’ compliance to controllers is difficult to estimate and model, as these depend on social and economic factors among others (Sorensen 2000). We partially address the effects of this assumption by allowing users to select through a range of possible routes during the traffic assignment stage of our algorithm.

The following notation is used in the remainder of this chapter:
The overall objective of this model is to ensure that the maximum number of evacuees from a given area complete their journeys within the course of a finite time horizon. The level of risk to which the evacuees are subjected to is assumed to increase with time, and we, therefore, seek to ensure that the journeys are completed as early as possible, using elapsed time as a surrogate for risk level. We adopt a composite function (1) consisting of two components that reflect these two objectives (1.1-1.2):

\begin{equation}
\begin{align*}
\text{maximise } Z &= C - \gamma W \\
C &= \sum_{t \in T} \left( \frac{1}{t} \sum_{i \in \mathcal{I}} N_{up,i}^t \right) \\
W &= \sum_{t \in T} \left( \sum_{s \in \mathcal{S}} N_{down,s}^t - \sum_{i \in \mathcal{I}} N_{up,i}^t \right)
\end{align*}
\end{equation}

Equation (1.1) penalises each departure in accordance with the time step at which they are due to complete their journeys. The \( N_{up,i}^t \) variable is used to capture the cumulative number of vehicle departures at every time step \( t \). The second component of the objective (1.2) seeks to minimise the total wait time \( W \), which captures the total duration of aggregate vehicle presence in the network (which includes delays). A weight factor \( \gamma \) is used to control the priority allocated to each of the two objectives. Given a set of nodes \( \mathcal{V} \) and links \( \mathcal{A} \), we create a network represented by a directed graph \( G(\mathcal{V}, \mathcal{A}) \). A maximum time horizon, \( T \), is defined along with discretised time step \( \Delta t \). The model formulation is as follows:

Objective:

\begin{equation}
\begin{align*}
\text{maximise } Z &= C - \gamma W \\
C &= \sum_{t \in T} \left( \frac{1}{t} \sum_{i \in \mathcal{I}} N_{up,i}^t \right) \\
W &= \sum_{t \in T} \left( \sum_{s \in \mathcal{S}} N_{down,s}^t - \sum_{i \in \mathcal{I}} N_{up,i}^t \right)
\end{align*}
\end{equation}

Subject to:
Effect of congestion. The linear problem is
CPLEX serves as an input for the traffic simulator, which updates
ulation that is solved in tandem with the
of computationally efficient soluti
2007; Li et al. 2010)
increasing the problem size to impractical levels for large transport networks
common approach consists on
The development of a fully linear mathematical
2.2.

\[ N_{\text{up},l}^t = \Delta t \sum_{i=1}^{t} q_{\text{in},i}^l \quad \text{if } l_i \in \mathcal{A}, \quad t \in T \]  
\[ N_{\text{down},l}^t = \Delta t \sum_{i=1}^{t} q_{\text{out},i}^l \quad \text{if } l_i \in \mathcal{A}, \quad t \in T \]  
\[ N_{\text{up},p}^t = N_{\text{down},p}^{t-\Delta t_{p,t}} \quad \text{if } p \in \mathcal{P}, \quad t \in T \]
\[ q_{\text{out},j}^t = \sum_{p \in \mathcal{O}^s} a_p^t R_p^t \quad \text{if } p \in \mathcal{O}^s, \quad t \in T \]
\[ V \leq R_p^t \leq U \quad \text{if } p \in \mathcal{O}^s, \quad t \in T \]
\[ \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} Y_{p,t} R_p^t \leq B_s \quad \text{if } p \in \mathcal{O}^s \]
\[ a_p^t = \{0,1\} \quad \text{if } p \in \mathcal{O}^s, \quad t \in T \]
\[ \sum_{p \in \mathcal{O}^s} a_p^t = K \quad \text{if } t \in T \]

Constraints (2.1) and (2.2) calculate the number of vehicles in every link \( l_i \) among the arcs \( \mathcal{A} \) of the network. Constraint (2.3) ensures that vehicles entering path \( p \) will arrive at the sink node within \( \Delta t_{p,t} \) time. Constraint (2.4) calculates the vehicular flow entering the network at time step \( t \) given an entering rate \( R_p^t \) on the activated paths \( a_p^t \). Equation (2.5) ensures that the path rate is bounded by a lower limit \( V \) and an upper limit \( U \) at every time step \( t \). Constraint (2.6) provides an upper bound for departures throughout the time horizon \( T \) that equates to the original demand. Finally, (2.7) and (2.8) state that \( a_p^t \) is a Boolean variable and that the paths activated must be limited to \( K \) for each origin, respectively.

The key limitation of the above formulation is its inability to determine \( \Delta t_{p,t} \), which is assumed to vary in accordance with vehicle flows. Therefore, the value of \( \Delta t_{p,t} \) is treated as a dynamic variable with a non-linear relationship with the decision variables \( R_p^t \) and \( a_p^t \) is non-linear. Chapter 4 describes a dynamic traffic assignment model that is used to evaluate \( \Delta t_{p,t} \) values for each time step.

2.2. LINEAR MODEL FORMULATION

The development of a fully linear mathematical model requires the linearization of traffic assignment. The common approach consists on duplicating the network for every discrete time-step of a time horizon \( T \), increasing the problem size to impractical levels for large transport networks (Lu et al. 2005; Shen et al. 2007; Li et al. 2010). Mathematical simplifications including no-overtaking (First-In-First-Out) and traffic holding require the inclusion of additional variables and non-convex constraints that preclude the adoption of computationally efficient solution algorithms (Doan & Ukkusuri 2012).

Given the limitations of linear dynamic traffic algorithms, we develop a linear mathematical formulation that is solved in tandem with the dynamic traffic simulator. The evacuation strategy developed using CPLEX serves as an input for the traffic simulator, which updates path travel times, \( \Delta t_{p,t} \), considering the effects of congestion. The linear problem is then solved using the updated network properties, creating a
deviation between the solution before and after $\Delta t_{p,t}$ is updated. This iterative process runs until the variation between solutions is negligible, thus estimating the effects of congestion within the constraints of the linear formulation. The main limitation of the proposed approach is the use of a static $\Delta t_{p,t}$ throughout the mission horizon for each iteration. We present the linear formulation, please refer to the notation description in Chapter 2 and the following parameters:

Parameters:

$P_{p,s}$ source $s$ belongs to path $p$ [-]  
$\Delta_p$ travel time of path $p$ [s]

Decision Variables:

$b_{p,t}$ slack variable to linearise relationship between $a_{p,t}$ and $N_{d_{p,t}}$ [veh/s]  
$N_{d_{p,t}}$ vehicle departures from path $p$ at time-step $t$ [-]  
$N_{a_{p,t}}$ vehicle arrival from path $p$ at time-step $t$ [-]

Objective:

$$\text{maximise } Z = C - \gamma W$$

$$C = \sum_{t \in T} \left( \frac{1}{t} \sum_{p \in P} N_{a_{p,t}} \right)$$

$$W = \sum_{t \in T} \left( \sum_{p \in P} N_{d_{p,t}} - \sum_{p \in P} N_{a_{p,t}} \right)$$

Subject to:

$$N_{a_{p,t}} = N_{d_{p,t}} - \Delta_{p,t}$$  
$p \in \mathcal{P}, t \in T$ (4.1)

$$R_{p,t} \Delta t_s = N_{d_{p,t}}$$  
$p \in \mathcal{P}, t \in T$ (4.2)

$$a_{p,t} V \leq R_{p,t} \leq a_{p,t} U$$  
$p \in \mathcal{P}, t \in T$ (4.3)

$$b_{p,t} - V(1 - a_{p,t}) \leq R_{p,t} \leq b_{p,t} - U(1 - a_{p,t})$$  
$p \in \mathcal{P}, t \in T$ (4.4)

$$B_s = \sum_{t \in T} \sum_{p \in P} P_{p,s} R_{p,t}$$  
$s \in \mathcal{S}$ (4.5)

$$\sum_{p \in P} a_{p,t} P_{p,s} = K$$  
$s \in \mathcal{S}, t \in T$ (4.6)

$$a_{p,t} = \{0,1\}$$  
$p \in \mathcal{P}, t \in T$ (4.7)
The objective function (3) is analogous to equation (2). Equation (4.1) ensures that vehicle equilibrium is satisfied within a path $p$ given an expected travel time $\Delta_{p,t}$. Equation (4.2) considers the departing flows $R_{p,t}$ and the total vehicles entering a path $N_{p,t}$. Constraints (4.3) and (4.4) ensure that the departing vehicles $R_{p,t}$ are bounded to $V$ and $U$ respectively while $R_{p,t} = 0$ for non-selected paths. Equation (4.5) equates the total demand in each source node $s$ to the total departures of each path, where $P_{p,s}$ equates to 1 when source $s$ belongs to path $p$ and 0 otherwise. Constraint (4.6) ensures that the total number of paths activated $a_{p,t}$ for each source node equals $K$. Finally, equations (4.7) and (4.8) bound the decision variables $a_{p,t}$ and $b_{p,t}$ to a Boolean and to 0 and $U$ respectively.

3. Travel Demand Model

This chapter presents a model that addresses the inability of the above formulation to capture the effects of congestion on vehicular flow, using a continuous time link-based kinematic wave model that was initially developed by Han et al. (2016). The latter considers the effects of flow propagation and congestion, including physical queuing and vehicle spillback, based on a reformulation of the scalar conservation law model (Lighthill & Whitham 1955; Richards 1956), the Hamilton-Jacobi equation and variational theory. Such a formulation has been independently derived by Jin (2015) and Han et al. (2016) and, when discretised, leads to the well-known link transmission model (Yperman et al. 2005). The model is path-based, with path selection and network properties provided previously by the user.

The traffic flow model consists of three components, a junction model, a source model and a link model. The junction model can be used to implement traffic signalling in the network. The source model is used to provide departure rates from specific nodes over the time-horizon of the evacuation. Finally, the link model is based on the fundamental triangular diagram and the concepts of demand and supply (Lebacque & Khoshyaran 1999), allowing the simulation to calculate in-flows and out-flows of links that are adjacent to the same junction and thus simulate shockwave propagation and vehicle spillback.

The traffic simulator is described by a system of time-continuous differential algebraic equations (DAE). It is also referred to as the dynamic network loading procedure in the dynamic traffic assignment literature. The underlying mathematical model is presented below following the notations introduced in Section Error! Reference source not found. - the reader is referred to Han et al. (2016) for its derivation and computational examples.

Parameters
- $C_i$: flow capacity of link $i$
- $L_i$: length of link $i$
- $w_i$: backward wave speed of link $i$,
- $\rho_i^{jam}$: jam density of link $i$
- $v_i$: forward wave speed of link $i$

Variables
- $D_i(t)$: demand of link $i$
- $q_{in,i}(t)$: inflow of link $i$
- $N_{up,i}(t)$: cumulative link entry volume
- $\mu_i^p(t,x)$: path disaggregation variable
- $a_{ij}(t)$: flow split at junction $j$
- $S_j(t)$: supply of link $j$
- $q_{out,i}(t)$: outflow of link $i$
- $N_{down,i}(t)$: cumulative link exit volume
- $\tau_i(t)$: link entry time function
- $\rho_i^{jam}$: jam density [$veh/m$]
The definitions of demand and supply that are used in this model conform with those proposed by Lebacque & Khoshyaran (1999), and correspond to the inflow and outflow capacity of a link. \( N_{\text{up},i}(t) \) and \( N_{\text{down},i}(t) \) denote the cumulative number of vehicles that have entered and exited the link \( i \) by time \( t \), respectively. In order to explicitly incorporate traveller route choices, the path disaggregate variable \( \mu^p_i(t, x) \) represents the proportion of traffic at point \( (t, x) \) on link \( i \) that is associated with path \( p \), where \( i \in p \). Based on \( N_{\text{up},i}(t) \) and \( N_{\text{down},i}(t) \), the link entry time function can be calculated based on the following flow propagation identity:

\[
N_{\text{up},i}(\tau_i(t)) = N_{\text{down},i}(t),
\]

where \( t \) denotes the link exit time. Finally, the flow split variable \( \alpha^i_j(t) \) indicate the probability of traffic exiting upstream link \( i \) that are heading towards downstream link \( j \), where both \( i \) and \( j \) are adjacent to junction \( J \). Given the above, and the formulation provided in the previous chapter, the DAE system for the dynamic network loading model is as follows:

\[
D_i(t) = \begin{cases} 
q_{\text{in},i}(t - \frac{L_i}{v_i}) & \text{if } N_{\text{up},i}(t - \frac{L_i}{v_i}) = N_{\text{down},i}(t) \\
C_i & \text{if } N_{\text{up},i}(t - \frac{L_i}{v_i}) > N_{\text{down},i}(t) 
\end{cases} \quad i \in \mathcal{A} \quad (5.1)
\]

\[
S_i(t) = \begin{cases} 
q_{\text{out},i}(t - \frac{L_i}{w_i}) & \text{if } N_{\text{up},i}(t) = N_{\text{down},i}(t - \frac{L_i}{w_i}) + \rho_j \text{ jam } L_j \\
C_i & \text{if } N_{\text{up},i}(t) < N_{\text{down},i}(t - \frac{L_i}{w_i}) + \rho_j \text{ jam } L_j 
\end{cases} \quad i \in \mathcal{A} \quad (5.2)
\]

\[
N_{\text{down},i}(t) = N_{\text{up},i}(\tau_i(t)) 
\quad i \in \mathcal{A} \quad (5.3)
\]

\[
\mu^p_i(t, 0) = \frac{q_{\text{out},i}(t, 0) \mu^p_i(\tau_i(t), 0)}{q_{\text{in},i}^p(t)} \quad \forall p \text{ such that } \{i, j\} \in p \quad (5.4)
\]

\[
A^i = \{\alpha^i_j(t)\}, \quad \alpha^i_j(t) = \sum_{p \in \mathcal{A}} \mu^p_i(\tau_i(t), 0) \quad i \in \mathcal{I}^i, j \in \mathcal{O}^i \quad (5.5)
\]

\[
|q_{\text{in},i}^k(t) = RS_k^i \left[ (D_i(t))_{i \in \mathcal{I}^i}, (S_i(t))_{j \in \mathcal{O}^i} \right] \quad k \in \mathcal{O}^i \quad (5.6)
\]

\[
|q_{\text{out},i}^l(t) = RS_l^i \left[ (D_i(t))_{i \in \mathcal{I}^i}, (S_i(t))_{j \in \mathcal{O}^i} \right] \quad l \in \mathcal{I}^j \quad (5.7)
\]

\[
\frac{d}{dt} N_{\text{up},i}(t) = q_{\text{in},i}(t), \quad \frac{d}{dt} N_{\text{down},i}(t) = q_{\text{out},i}(t) \quad i \in \mathcal{A} \quad (5.8)
\]

Equations (5.1)-(5.2) define the link demand and supply, given the cumulative entry and exit volumes at earlier times. The relationship between entry and exit times at every link, based on flow conservation and the First-In-First-Out principle is defined by equation (5.3). Equation (5.4) propagates the path disaggregation variable between consecutive links, while equation (5.5) states that the flow distribution at a junction with multiple downstream links can be uniquely determined by the relevant path disaggregation variables. Equations (5.6)-(5.7) determine the inflow and outflow of links that are adjacent incident to a junction \( J \) based on the Riemann Solver (Garavello et al. 2016), which depends on the junction topology and specific traffic management scenarios such as signal control and priorities. Finally, (5.8) links the two types of fundamental flow variables: flow and cumulative volume. This DAE system can be solved in discrete time in a forward fashion; see Garavello et al. (2016) for computational examples of this system in a number of traffic networks.

The traffic model uses a set of paths as principal inputs. While safety concerns in real-live evacuation exercises may condition routing, we assume that for the purposes of this study a reduction of vehicle
exposure in the network will increase user safety. This simplification allows us to use a k-shortest path algorithm based on Yen’s algorithm (Yen 1971) to generate a set of multiple route choices to the exit node. Our implementation of the evacuation routing algorithm is detailed in the Pseudocode 1 below, and incorporates a Dijkstra’s-based solver (Dijkstra 1959) to determine individual paths.

```
1. Algorithm Evacuation Routing is
2.   Input: graph G, origins O_v, destinations D_v, number of paths K
3.   Output: shortest paths L_p
4.   spurPath p_v → CreateEmptyList()
5.   p* ← Dijkstra(G, O_v, D_v)
6.   for each path p in L_p:
7.       for each vertex v in p
8.           spur vertex v_s ← v
9.           root path p_r ← p[0,v]
10.      for each subPath p̅ in L_p
11.         if p_r == p̅[0,v]
12.            G. RemoveLink (p̅[v,v + 1]
13.      for each pathNode v_p in p_r except v_s
14.          G. RemoveVertex (p_r[v_p])
15.          p̅_v ← Dijkstra(G, O_v, D_v)
16.          newPath p_n = p_r + p̅_v
17.          spur path list L_p_v ← p_n
18.      G. Restore()
19.      if L_p_v is empty
20.         break
21.   L_p_v. SortByCost()
22.   L_p ← L_p_v. First()
23.   L_p_v. RemoveFirst()
24.   return L_p
```

Pseudocode 1 Shortest path algorithm pseudocode.

3.1. INTEGRATION
We integrate both models using a feedback loop as presented in Pseudocode 2. The traffic model requests the initial paths and network conditions, a simulation time-step Δt_s and decision time-step Δt_d as inputs. Between each Δt_s the model updates the network properties based on current vehicular flow, maintaining network properties constant during Δt_s. The traffic model outputs the current network conditions and calls the evacuation routing model after each Δt_d, returning the new shortest paths accounting for congestion. A list of paths is updated in case the returned shortest paths have not been previously explored. The cost of all paths in the list is updated as the algorithm runs for traceability purposes.

The proposed traffic assignment model has been applied to a range of network sizes and configurations to determine its scalability. Networks A, B, C and D are defined in the Appendix, while the Sioux-Falls and Enriched Sioux Falls networks are presented in further detail in Chapter 4.2. Different junction configurations are tested to determine the effect in computational time. As shown in Table 3, the traffic assignment model can operate in complex networks under reasonable runtime. The computation time increases with network size and network complexity.

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Links</th>
<th>Origins</th>
<th>Destinations</th>
<th>Junctions</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>D</td>
<td>0.91</td>
</tr>
</tbody>
</table>
Table Legend
Junction Type: D – Diverging, C – Converging, M – Multiple junction structures.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>282</td>
<td>334</td>
<td>21</td>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td>ESF</td>
<td>24</td>
<td>76</td>
<td>4</td>
<td>1</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>76</td>
<td>4</td>
<td>1</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>D-C</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

FORMULATION OF OPTIMAL EVACUATION STRATEGY MODEL

In this chapter, we introduce and evaluate the bi-level evacuation optimisation model. We present the final mathematical formulation to evaluate the effectiveness of the evacuation scheme as a combination of equation sets (2) and (5):

Objective:

\[
\text{maximise } Z = C - \gamma W
\]

\[
C = \sum_{t \in T} \left( \frac{1}{t} \sum_{i \in \mathcal{D}} N^t_{up,i} \right) \quad d \in \mathcal{D} \tag{6.1}
\]

\[
W = \sum_{t \in T} \left( \sum_{j \in \mathcal{O}} N^t_{down,j} - \sum_{i \in \mathcal{D}} N^t_{up,i} \right) \quad d \in \mathcal{D}, s \in \mathcal{S} \tag{6.2}
\]

Subject to:

\[
N^t_{up,i} = \Delta t_s \sum_{i=1}^{t} q^t_{in,i} \quad l_i \in \mathcal{A}, t \in T \tag{7.1}
\]

\[
N^t_{down,i} = \Delta t_s \sum_{i=1}^{t} q^t_{out,i} \quad l_i \in \mathcal{A}, t \in T \tag{7.2}
\]

Pseudocode 2 Integrated traffic model.

4. FORMULATION OF OPTIMAL EVACUATION STRATEGY MODEL

In this chapter, we introduce and evaluate the bi-level evacuation optimisation model. We present the final mathematical formulation to evaluate the effectiveness of the evacuation scheme as a combination of equation sets (2) and (5):

Objective:

\[
\text{maximise } Z = C - \gamma W
\]

\[
C = \sum_{t \in T} \left( \frac{1}{t} \sum_{i \in \mathcal{D}} N^t_{up,i} \right) \quad d \in \mathcal{D} \tag{6.1}
\]

\[
W = \sum_{t \in T} \left( \sum_{j \in \mathcal{O}} N^t_{down,j} - \sum_{i \in \mathcal{D}} N^t_{up,i} \right) \quad d \in \mathcal{D}, s \in \mathcal{S} \tag{6.2}
\]

Subject to:

\[
N^t_{up,i} = \Delta t_s \sum_{i=1}^{t} q^t_{in,i} \quad l_i \in \mathcal{A}, t \in T \tag{7.1}
\]

\[
N^t_{down,i} = \Delta t_s \sum_{i=1}^{t} q^t_{out,i} \quad l_i \in \mathcal{A}, t \in T \tag{7.2}
\]
The objective function presented in Equation (6) remains analogous to Equation (2). It aims to maximise the number of evacuees, promoting early evacuations, while reducing the time vehicles remain in the network. Many of the constraints are carried over with respect to Chapter Error! Reference source not found.. Equations (7.1) and (7.2) describe the calculation of total vehicles in every link \( L_i \) among the arcs of the network \( \mathcal{A} \). Constraint (7.2) determines vehicles entering the network at a defined rate \( R_p^t \). Constraints (7.9-7.10) propose lower and upper bounds on the departure rates based on available demand. Finally, the number of activated paths is constrained by Equations (7.11-7.12).

The remaining equations are carried over from the traffic model presented in Chapter 4.1. Equations (7.3-7.4) are analogous to (5.1-5.2) and define supply and demand for each link, and (7.5-7.8) reflect (5.4-5.7) which determine flow distribution through junctions using Riemann solvers. The final integrated method is provided in Pseudocode 3.
4.1. LINEAR FORMULATION PERFORMANCE

The linearised problem is solved using the set of simple networks provided in the Appendix Table 3. The key performance parameters explored include the fitness value as described in equation (6); the weighted cumulative evacuation, defined as $\sum_{t} N_{up,i,t}$ in the same equation; the total number of departures $\sum_{t} N_{down,i,t}$, which denotes the number of vehicles that are allowed to leave their residence; and the total number of finished evacuations $\sum_{t} N_{up,i,t}$. Finally, we compute the mean travel time for each evacuation strategy.

We present the results in Table 4. In most networks, the hybrid method outperforms the linear formulation regarding fitness value. The main limitation of the linearised problem is the assumption that travel times are fixed during the simulation. Despite the model being iteratively solved in conjunction with the traffic simulator until convergence, optimisation techniques that can consider congestion effects during its evaluation will provide a better optimum. Similar performance levels are only obtained in the smallest network A – we attribute this to its simple layout, which only contains a divergent junction with a small effect on congestion. However, as the complexity of the network increases the deviation between both solution increases. As an example, in network B which contains a single merging junction, there is a 30% difference in fitness values between both methods. Therefore, the linear formulation proposes an insufficient framework to evaluate and optimise evacuation strategies.

Table 4
Linear Model and Hybrid model comparison. SF denotes Sioux-Falls network.

<table>
<thead>
<tr>
<th>Solution Approach</th>
<th>Network</th>
<th>Fitness Value (vehicle / time step)</th>
<th>Weighted evacuations (vehicles / time step)</th>
<th>Total departures (vehicles)</th>
<th>Total evacuations (vehicles)</th>
<th>Mean travel time (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>A</td>
<td>2,623</td>
<td>2,632</td>
<td>3,064</td>
<td>2,969</td>
<td>36.3</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>2,879</td>
<td>2,914</td>
<td>3,159</td>
<td>2,789</td>
<td>166.4</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>2,793</td>
<td>2,819</td>
<td>3,710</td>
<td>3,295</td>
<td>151.9</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>3,501</td>
<td>3,685</td>
<td>6,709</td>
<td>4,268</td>
<td>252.7</td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>7,100</td>
<td>7,259</td>
<td>9,692</td>
<td>7,907</td>
<td>187.6</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>SF</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>2,613</td>
<td>4,113</td>
<td>3,134</td>
<td>4,193</td>
<td>9,092</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2,622</td>
<td>4,149</td>
<td>3,146</td>
<td>4,304</td>
<td>9,259</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3,064</td>
<td>4,908</td>
<td>3,710</td>
<td>6,929</td>
<td>11,394</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2,966</td>
<td>4,491</td>
<td>3,575</td>
<td>5,095</td>
<td>10,255</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35.8</td>
<td>158.5</td>
<td>49.8</td>
<td>187.5</td>
<td>187.6</td>
<td></td>
</tr>
<tr>
<td>Percentage difference</td>
<td>-0.4</td>
<td>30.0</td>
<td>10.9</td>
<td>16.5</td>
<td>21.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.4</td>
<td>29.8</td>
<td>10.4</td>
<td>14.4</td>
<td>21.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>35.6</td>
<td>0.0</td>
<td>3.2</td>
<td>14.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.1</td>
<td>37.9</td>
<td>7.8</td>
<td>16.2</td>
<td>22.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>4.7</td>
<td>67.2</td>
<td>25.8</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

4.2. Test Network

The methodology is applied to the Sioux-Falls network (Bar-Gera 2016), consisting of 24 nodes and 76 arcs. A danger zone is specified to cover nodes 10, 11, 14 and 15, and a single evacuation point is specified in node 1 as shown in Figure 1. The model aims to evacuate users generated in the danger zone and take them to the safe node. All nodes within the evacuation area are assumed to have an equal pre-defined demand would flood the network in case of simultaneous departure. A large-scale version of the Sioux-Falls network is used to determine its scalability and applicability. The selected network consists of 282 nodes and 334 arcs (Chakirov & Fourie 2014). A small alteration to the evacuation procedure is introduced to compensate for the larger network size – we introduce secondary sink node in node 2. It is important to note that, due to the new network structure, the source nodes increase from 4 nodes (Figure 1) to 20 to create the same danger zone. For the remainder of this paper, we will refer to this network as Enriched Sioux Falls (ESF).

We propose a selection of Derivative-Free Algorithms (DFO) to evaluate the performance of each in Table 5. Similar to simulation-optimisation strategies algorithms, DFOs are categorised into local greedy and global elitist algorithms. Within local search, direct methods continuously evaluate trial solutions, and model-based search creates a surrogate model to perform guided search. Global deterministic and model-based search are analogous to the local direct and model-based search counterparts but partition the search space. Finally, stochastic methods allow intermediate lower quality solutions to avoid local optima.

We consider both local and global methods in our test, including deterministic and stochastic methods. A scenario with 2 activated paths and a decision time step size $\Delta t_d$ of 10 seconds was selected to test all algorithms, using the Sioux Falls network presented. As shown in Table 6, both local methods provide far from optimal solutions due to their inability to escape local optima. On the performance of the global algorithms, the Simulated Annealing and the Genetic Algorithm outperform other solvers. The latter provides the best solution, with the greatest number of departures, evacuation and shortest mean travel time.

Given the results of this analysis (Table 6), we chose to adopt the Genetic Algorithm as the optimisation solver used by the hybrid approach. This algorithm consists of three main components: selection, crossover and mutation. We use a binary tournament to pair solutions and select the best as indicated by their fitness value. Although this process discards well-performing solutions as does not compare candidates to the population, it can be parallelised to improve computational performance (Noraini & Geraghty 2011).

Crossover combines two parent solutions to create two new unique solutions. We use a BLX-alpha crossover method as it allows exploration of solution further than the parents (Eshelman & Schaffer 1993). The size of the search space explored is defined by parameter $\alpha$, where a greater value increases the search space. Finally, mutation alters the parent solutions to create unique solutions outside the range of the parent solutions. We use the non-uniform mutation (Michalewicz 1992), which reduces the mutation range after every generation, encouraging convergence as the algorithm runs. The decrease in range rate is controlled by the $\beta$ parameter, where a larger value of $\beta$ increases the rate of decrease.
Figure 1 Sioux Falls network diagram.

Figure 2 Enriched Sioux Falls network diagram.
Table 5
Derivative-Free Optimisation algorithms used.

<table>
<thead>
<tr>
<th>DFO</th>
<th>Algorithm Type</th>
<th>Solver</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nelder-Mead Simplex</td>
<td>Local Direct</td>
<td>fminsearch</td>
<td>Mathworks</td>
</tr>
<tr>
<td>Implicit Filtering</td>
<td>Local Model-Based</td>
<td>imfil</td>
<td>Kelley (2011)</td>
</tr>
<tr>
<td>Branch and Fit</td>
<td>Global Deterministic</td>
<td>snobfit</td>
<td>Huyer &amp; Neumaier (2008)</td>
</tr>
<tr>
<td>Particle Swarm</td>
<td>Global Stochastic</td>
<td>Pswarm</td>
<td>Vaz &amp; Vicente (2007)</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>Global Stochastic</td>
<td>Original</td>
<td>-</td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>Global Stochastic</td>
<td>Original</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6
Derivative-Free Optimisation techniques key performance indicator comparison.

<table>
<thead>
<tr>
<th>DFO</th>
<th>Fitness Value (vehicle / time step)</th>
<th>Weighted evacuations (vehicles/ time step)</th>
<th>Total departures (vehicles)</th>
<th>Total evacuations (vehicles)</th>
<th>Mean travel time (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM</td>
<td>1,860</td>
<td>1,933</td>
<td>2,364</td>
<td>1,624</td>
<td>285.8</td>
</tr>
<tr>
<td>IF</td>
<td>4,627</td>
<td>4,774</td>
<td>6,308</td>
<td>4,686</td>
<td>235.8</td>
</tr>
<tr>
<td>BF</td>
<td>8,308</td>
<td>8,478</td>
<td>10,836</td>
<td>9,438</td>
<td>190.2</td>
</tr>
<tr>
<td>PS</td>
<td>8,428</td>
<td>9,316</td>
<td>11,182</td>
<td>9,415</td>
<td>191.3</td>
</tr>
<tr>
<td>SA</td>
<td>8,925</td>
<td>9,130</td>
<td>10,167</td>
<td>9,611</td>
<td>189.2</td>
</tr>
<tr>
<td>GA</td>
<td>9,092</td>
<td>9,259</td>
<td>11,393</td>
<td>10,254</td>
<td>187.6</td>
</tr>
</tbody>
</table>

4.3. Parameter Tuning
We proceed with manual tuning to determine suitable control parameters for the genetic algorithm. The control parameters considered include $\alpha$ and $\beta$, which control crossover and mutation search spaces, and $p_c$ and $p_m$, which represent probability parameters for crossover and mutation respectively. A total of 9 scenarios are considered, with their parameters summarised in Table 7. The evacuation parameters decision time step $\Delta t_d$, and activated paths $K$ remain fixed in all cases. The scenario naming convention describes the evacuation parameters used such that K1 denotes $K = 1$ and D10 signifies $\Delta t_d = 10$.

Table 7
Sioux Falls evacuation calibration scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>K</th>
<th>$\Delta t_d$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$p_c$</th>
<th>$p_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>K2D10</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-a1</td>
<td>2</td>
<td>10</td>
<td></td>
<td>0.6</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-a2</td>
<td>2</td>
<td>10</td>
<td>0.8</td>
<td>0.5</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-b1</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.6</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-b2</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.8</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-c1</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
<td>0.6</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-c2</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>K2D10-m1</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>K2D10-m2</td>
<td>2</td>
<td>10</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Table 8
Parameter calibration results for the Genetic Algorithm.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fitness Value (vehicle / time step)</th>
<th>Weighted evacuations (vehicles/time step)</th>
<th>Total departures (vehicles)</th>
<th>Total evacuations (vehicles)</th>
<th>Mean travel time (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K2D10</td>
<td>9,092</td>
<td>9,259</td>
<td>11,393</td>
<td>10,254</td>
<td>187.6</td>
</tr>
<tr>
<td>K2D10-a1</td>
<td>9,214</td>
<td>9,380</td>
<td>11,659</td>
<td>10,327</td>
<td>183.2</td>
</tr>
<tr>
<td>K2D10-a2</td>
<td>9,213</td>
<td>9,380</td>
<td>11,640</td>
<td>10,308</td>
<td>181.1</td>
</tr>
<tr>
<td>K2D10-b1</td>
<td>9,213</td>
<td>9,380</td>
<td>11,605</td>
<td>10,333</td>
<td>157.9</td>
</tr>
<tr>
<td>K2D10-b2</td>
<td>9,287</td>
<td>9,340</td>
<td>10,321</td>
<td>10,301</td>
<td>181.1</td>
</tr>
<tr>
<td>K2D10-c1</td>
<td>9,211</td>
<td>9,387</td>
<td>11,106</td>
<td>10,243</td>
<td>191.5</td>
</tr>
<tr>
<td>K2D10-c2</td>
<td>9,174</td>
<td>9,358</td>
<td>11,786</td>
<td>10,340</td>
<td>188.1</td>
</tr>
<tr>
<td>K2D10-m1</td>
<td>9,285</td>
<td>9,379</td>
<td>10,411</td>
<td>10,305</td>
<td>161.4</td>
</tr>
<tr>
<td>K2D10-m2</td>
<td>9,011</td>
<td>9,197</td>
<td>12,298</td>
<td>10,237</td>
<td>186.9</td>
</tr>
</tbody>
</table>

As observed in Table 8, higher values of $\alpha$ and $\beta$ provide better performance than the original best performing scenario according to the fitness value is the original K2D10-b2. Concerning $\alpha$, both scenarios with higher $\alpha$ yield better fitness values. Both provide a large search space initially which gradually decreases in later iterations. On the operator probabilities, increasing crossover rate also improves the performance, as well as reducing the mutation rate. The first provides more solution combinations, while the second reduces the stochasticity of solutions. Despite mutation providing a means to deviate from local optima, high mutation probabilities may result in a high-degree stochastic search algorithm rather than elitist search mechanism. Conversely, providing a very low value does not allow the algorithm to explore sufficient area of the solution surface to find a close to the optimal solution, which is the case in m2.

4.4. RESULTS
We review the solutions for the scenarios presented in Table 9. A benchmark problem is used to simulate a worst-case solution in which all demand is released simultaneously at the beginning of the evacuation.

Table 9
Sioux Falls evacuation calibration scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$k$</th>
<th>$\Delta t_d$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$p_c$</th>
<th>$p_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark</td>
<td>5</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>K1D10</td>
<td>1</td>
<td>10</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
<tr>
<td>K1D20</td>
<td>1</td>
<td>20</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
<tr>
<td>K2D10</td>
<td>2</td>
<td>10</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
<tr>
<td>K2D20</td>
<td>2</td>
<td>20</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
<tr>
<td>K3D10</td>
<td>3</td>
<td>10</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
<tr>
<td>K3D20</td>
<td>3</td>
<td>20</td>
<td>0.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.05</td>
</tr>
</tbody>
</table>

The model was run on a modest workstation with an Intel Core i7-4790 CPU and 8GB RAM. The runtime increases given a higher value of $k$, but the variation is not substantial, as all the scenarios required approximately 5 and 16 hours to complete in the SF and the ESF respectively. This level of performance was deemed to be acceptable for pre-planning purposes yet unsuitable for real-time implementation. Further efforts to accelerate the algorithm are to increase parallelisation of the genetic algorithm or the use of high-performance cloud computing facilities.
The results are summarised in Table 10. The benchmark scenario involves an uncoordinated evacuation which saturates the network immediately after the evacuation warning is emitted. As expected, this is the worst performing problem instance - the evacuation process is not able to recover from the immediate surge in demand, with rate of vehicle arrivals at the evacuation point decreasing over time (Figure 3). This scenario also yields the lowest total departures and evacuations among all scenarios, meaning that congestion occurs close to the source nodes, preventing vehicles from accessing the network.

Other instances performed better due to a combination of reduced decision time-steps and increased activated route paths. The provision of additional paths increases the network capacity, thus allowing a higher rate of evacuations to take place. However, the evacuation improvement is reduced between $K = 3$ and $K = 2$ as a result of the limited accessibility of node 1. Only two links can access the sink node, causing $K > 2$ paths to converge, resulting in congestion at the junctions.

Another important observation relate to the constant evacuation rate when $t > 100$, as observed in Figure 3. Therefore, the solutions not only significantly improve upon a do-nothing approach, but also create a stable evacuation process that allows users to exit the network at a constant rate. Interestingly, the do-nothing solution performs similarly to the rest at the beginning of the evacuation, under free-flow network conditions. Evacuation rate promptly decreases as congestion starts affecting the evacuation process.

On the performance of the genetic algorithm, Figure 4 shows that all the solutions converge early, finding near-optimal solutions in generation 100 and improving the final solution in small increments thereafter.

### Table 10
Sioux Falls network results.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fitness Value</th>
<th>Weighted evacuations (vehicles)</th>
<th>Total departures (vehicles)</th>
<th>Total evacuations (vehicles)</th>
<th>Mean travel time (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value %</td>
<td>Value %</td>
<td>Value %</td>
<td>Value %</td>
<td>Value %</td>
</tr>
<tr>
<td>Benchmark</td>
<td>5,341 0</td>
<td>5,458 0</td>
<td>5,991 0</td>
<td>4,934 0</td>
<td>204.6 0</td>
</tr>
<tr>
<td>K1D10</td>
<td>8,664 62</td>
<td>8,831 62</td>
<td>11,302 89</td>
<td>9,771 98</td>
<td>192.1 6</td>
</tr>
<tr>
<td>K1D20</td>
<td>8,554 60</td>
<td>8,723 60</td>
<td>11,595 94</td>
<td>9,878 100</td>
<td>182.2 11</td>
</tr>
<tr>
<td>K2D10</td>
<td>9,091 70</td>
<td>9,259 70</td>
<td>11,393 90</td>
<td>10,254 108</td>
<td>187.6 8</td>
</tr>
<tr>
<td>K2D20</td>
<td>8,994 68</td>
<td>9,157 68</td>
<td>12,298 105</td>
<td>10,156 106</td>
<td>184.3 10</td>
</tr>
<tr>
<td>K3D10</td>
<td>9,199 72</td>
<td>9,347 71</td>
<td>11,679 95</td>
<td>10,150 106</td>
<td>180.6 12</td>
</tr>
<tr>
<td>K3D20</td>
<td>9,172 72</td>
<td>9,341 71</td>
<td>11,570 93</td>
<td>10,308 109</td>
<td>181.3 11</td>
</tr>
</tbody>
</table>
Figure 3 Sioux Falls network – Weighted sum of evacuees over each time step for every scenario considered.

Figure 4 Sioux Falls network – Evolving fitness value over each generation.

Table 11
Enriched Sioux Falls network results.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fitness Value</th>
<th>Weighted evacuations (vehicles)</th>
<th>Total departures (vehicles)</th>
<th>Total evacuations (vehicles)</th>
<th>Mean travel time (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>%</td>
<td>Value</td>
<td>%</td>
<td>Value</td>
</tr>
<tr>
<td>Benchmark</td>
<td>1,233</td>
<td>0</td>
<td>1,517</td>
<td>0</td>
<td>3,920</td>
</tr>
<tr>
<td>K1D10</td>
<td>1,949</td>
<td>58</td>
<td>2,321</td>
<td>53</td>
<td>8,025</td>
</tr>
<tr>
<td>K1D20</td>
<td>1,933</td>
<td>57</td>
<td>2,273</td>
<td>50</td>
<td>7,495</td>
</tr>
<tr>
<td>K2D10</td>
<td>2,185</td>
<td>77</td>
<td>2,534</td>
<td>67</td>
<td>6,856</td>
</tr>
<tr>
<td>K2D20</td>
<td>2,187</td>
<td>77</td>
<td>2,521</td>
<td>66</td>
<td>6,819</td>
</tr>
<tr>
<td>K3D10</td>
<td>2,269</td>
<td>84</td>
<td>2,572</td>
<td>70</td>
<td>6,185</td>
</tr>
<tr>
<td>K3D20</td>
<td>2,234</td>
<td>81</td>
<td>2,546</td>
<td>68</td>
<td>6,346</td>
</tr>
</tbody>
</table>
The algorithm is found to perform better in the Enriched Sioux Falls network case study, where we observe an improvement in the fitness value when increasing the number of activated paths and when reducing the decision time step. The results suggest that demand management is an effective method to optimise evacuations as all solutions improve evacuation upon the benchmark scenario and create stable evacuation processes. The behaviour is represented by the constant gradient in Figure 3 and Figure 5. It is interesting to note that the benchmark solution significantly reduces in effectiveness as the evacuation is carried out, which may be particularly misleading for coordinating agencies on the effectiveness of the evacuation. Referring to Figure 5, it is observed that the weighted evacuation indicator is higher for the benchmark scenario than for the K1 scenarios at the beginning of the evacuation.

Given the total departures and evacuations presented in Table 10 the process in the SF is more successful than in the ESF. More vehicles successfully evacuated the SF, and the difference between total departures and evacuations is larger in the ESF case. This is a result of the higher accessibility of the source nodes in the SF network. By inspecting Figure 2, most source nodes in the ESF are interconnected, resulting in congestion when vehicles travel to neighbour nodes. Another factor to account for is the small improvement in performance increasing paths: in both scenarios, the provision of additional paths leads to reduced performance gains. The demand monitoring scheme provides a greater evacuation improvement than the subsequent path provision. In both scenarios, the K1 scenario improved the initial scheme by ~60%, and adding subsequent paths resulted in a 10% improvement in the SF case, and 25% in the ESF network. The variability suggests that higher path convergence exists in the ESF, and the slight performance increase is a result of the additional capacity.

Figures 4 and 6 indicate that in both networks the algorithm converge into a stable solution. As expected, problem instances with a larger number of activated paths require a longer time to converge, due to a higher number of decision variables that need to be considered. Potential solutions include the use of a modified version of the k-shortest path algorithm, or the consideration of alternative evacuation strategies that utilise crossing elimination, which would reduce network congestion.

**Figure 5** Enriched Sioux Falls network - Weighted sum of evacuees over each time step for every scenario considered. All solutions follow the same trend.
5. **CONCLUSION**

In this paper, we develop a bi-level modelling framework that is used to optimise evacuation strategies using a network demand management approach. A driving principle that was adopted in its design is that centralised control is difficult to apply in emergency evacuations, therefore giving users an increased level of control over their choice of paths, when compared to previous studies. An efficient traffic flow model is adopted, implemented as a black-box model that captures traffic behaviour patterns during evacuation.

The methodology was tested in two different networks under a variety of scenarios. The analysis of the results indicates that demand management is an effective method to improve the efficiency of evacuations – with improvements observed across all key performance indicators when compared to the benchmark case. A range of Derivative-Free Optimisation algorithms were evaluated, with Genetic Algorithms shown to outperform other options and adopted as the solver that underpins the overall solution search process.

Even though a degree of personal choice (for routing decisions) is considered by this framework, further research is essential to establish the impact of varying rates of departure time compliance and information quality. Our future research in this area will explore the applicability of probabilistic decision-making techniques, including the use of myopic routing models. Monte Carlo simulation can be used to model a range of user behaviours, which is expected to come at the expense of longer runtimes. Furthermore, the use of a variety of transport modes should be considered to aid population groups with accessibility problems or lack of access to private vehicles.

**ACKNOWLEDGEMENTS**

The research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) as part of the Sustainable Civil Engineering Centre for Doctoral Training (Grant number EP/L016826/1).

**REFERENCES**


Jin, W.L., 2015. Continuous formulations and analytical properties of the link transmission model. 
*Transportation Research Part B: Methodological*, 74, pp.88–103.


**APPENDIX**

We provide below the network configurations that were introduced in Chapter 4.1 for the evaluation of the formulation in in larger problem instances.

A.  

```
1  2  3  5
```

B.  

```
1  3  5  6
```

C.  

```
4  6  2  4
```