Geoinformatica (2021) 25:655–678
https://doi.org/10.1007/s10707-021-00435-0

Conflict-Free Evacuation Route Planning

Roxana Herschelman[1] · **Ahmad Qutbuddin**[1] · **KwangSoo Yang**[1]

Received: 17 August 2020 / Revised: 22 February 2021 / Accepted: 3 April 2021 /
Published online: 29 April 2021

Abstract

Given a transportation network, a population, and a set of destinations, the goal of Conflict-Free Evacuation Route Planning (CF-ERP) is to produce routes that minimize the evacuation time for the population with no spatiotemporal movement-conflicts. The CF-ERP problem is an essential component of civic emergency preparedness in the wake of man-made or natural disasters (e.g., terrorist acts, hurricanes, or nuclear accidents). This problem is challenging because of the large size of network data, the large number of evacuees, and the need to account for capacity constraints and the conflict-free constraint. Previous work has focused on minimizing the evacuation time on spatiotemporal networks. However, these approaches cannot minimize potential movement-conflicts that cause traffic accidents, congestion, and delays. We propose novel approaches for CF-ERP to meet the conflict-free constraint while minimizing the evacuation time for the population. Experiments using real-world datasets demonstrate that the proposed algorithms produce evacuation routes with no movement-conflicts and have comparable solution quality to related work.

Keywords Evacuation Route Planning · Node-independent route · Constrained network flow algorithm

1 Introduction

Given a transportation network with edge capacity constraints, initial node occupancy, and destination locations, the goal of Conflict-Free Evacuation Route Planning (CF-ERP) is to produce routes that minimize the evacuation time with no spatiotemporal movement-conflicts. We define a conflict as a situation where two routes arrive at the same intersection

✉ Roxana Herschelman
rohriniuc2014@fau.edu

Ahmad Qutbuddin
aqutbuddin2017@fau.edu

KwangSoo Yang
yangk@fau.edu

[1] Computer and Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, EE 428, Boca Raton, FL 33431-0991, USA

at the same time (e.g., crisscrossing and merging). Figure 1a shows an example input of CF-ERP consisting of a spatial graph with two source nodes (i.e., $S1$ and $S2$) and two destination nodes (i.e., $D1$ and $D2$). Every edge has a travel time and a capacity constraint. Assume that each source node has a unit population. Figure 1b–d show example outputs of evacuation route planning. Assume that every evacuee starts moving at a time-point of 0 toward a destination node. Figure 1b shows a crisscrossing conflict at a time-point of 2 between the two different routes, whereas Fig. 1c shows a merging conflict at a time-point of 2 between the two different routes. These conflict points cause intersection delays and increase the potential risks (e.g., vehicle turning-movement conflicts). Figure 1d shows one example output of CF-ERP that can minimize the evacuation time with no movement-conflicts.

CF-ERP is a critical component for identifying the safest and most efficient routes in the wake of man-made or natural disasters (e.g., terrorist acts, hurricanes, or nuclear accidents). The key constraint is to mitigate potential movement-conflicts that cause traffic accidents, congestion, and delay. This problem is challenging because of the large size of network data, the large number of evacuees, the capacity constraints of the network, and the conflict-free

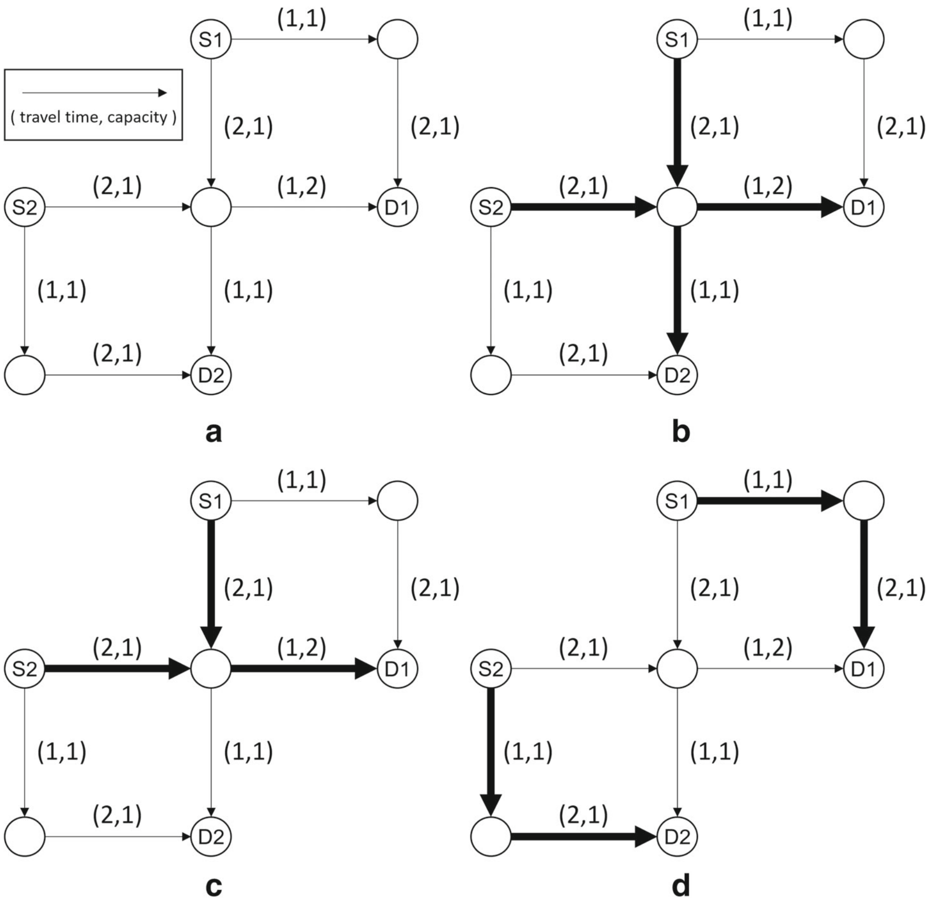


Fig. 1 Example input and outputs of evacuation route planning

constraint on routes. In this paper, we propose a novel algorithm for CF-ERP to meet the conflict-free constraint while minimizing the evacuation time for the population. The core idea of the proposed approach is to identify all possible node-independent shortest paths at each starting time-point and construct a flow graph that honors the capacity constraints of the network and the conflict-free constraint on routes.

1.1 Application domain

CF-ERP is important for a wide range of societal applications such as transportation management, logistics, public safety, and resource assignment. CF-ERP's objective is to minimize the evacuation time for all expected residents in the evacuation zone (EZ). This is important for reducing exposure to disasters and ensuring safety for evacuees. During the recent Hurricane Irma that hit in Florida, evacuation route computation was an essential component of civic emergency preparedness [28]. One major issue during this storm was a mass evacuation in which people blindly relied on Florida evacuation routes without considering potential risks (e.g., traffic congestion, movement-conflicts, etc.), causing major delays. It is well-known that traffic delays and incidents during evacuation occur most at intersections [27]. For example, during Hurricane Irma, most of the evacuees had experienced significant traffic congestion at the intersection of interstate 75 and Florida Turnpike (see Fig. 2).

CF-ERP removes movement conflict points in evacuation routes and reduces the interaction delays among vehicles. The importance of the conflict-free constraint is evident, in particular, for massive evacuation planning. The conflict-free constraint helps us identify safe and efficient routes and avoid the potential traffic congestion at intersections.

An algorithm for CF-ERP should also be fast and scalable enough to enhance its value under disaster conditions. For example, in emergency or disaster situations, planners must often respond to unforeseen changes. If an algorithm runs quickly, it could be rerun after each unforeseen change and can thus better adapt to specific circumstances (e.g., traffic accident or flood on a road segment) than a slower algorithm.

Fig. 2 Traffic backup during Hurricane Irma, on northbound interstate 75 at intersection with Florida turnpike. (September 7th, 2017). Courtesy USA Today [25]

1.2 Problem definition

In our problem formulation of the CF-ERP problem, a transportation network is represented and analyzed as a directed graph composed of nodes and edges. Every node represents a spatial location in geographic space (i.e., road intersections). Every edge between two nodes represents a road segment and has a travel time and a maximum capacity constraint. We assume that the transportation network follows the First-In First-Out (FIFO) principle. The $CF\text{-}ERP(N, E, C, T, S, D)$ problem is defined as follows:

Input: A transportation network G with

- a set of nodes N and a set of edges E,
- a set of positive integer capacities for edges $C : E \rightarrow Z^+$,
- a set of positive integer travel times for edges $T : E \rightarrow Z^+$,
- the total number of evacuees and their initial location ($S \subset N$), and
- a set of evacuation destinations $D \subset N$,

Output: An evacuation plan consisting of a set of origin-destination routes and a scheduling of evacuees on each route.

Objective:

- Minimize the evacuation time.

Constraints:

- No conflict between routes.
- The scheduling of evacuees on each route observes the capacity constraints.
- Edge travel time preserves FIFO (First-In-First Out) property.

Definition 1 Conflict: Given two different routes, a conflict can be defined when two routes arrive at the same intersection (i.e, node) at the exact same time (e.g., crisscrossing, merging, etc.). Conflicts raise the potential traffic risks (e.g., traffic delays and incidents) during evacuation.

Definition 2 First-In First-Out (FIFO): The FIFO principle assumes that a vehicle cannot overtake or pass another vehicle proceeding in the same direction. The FIFO principle guarantees that the conflict can be realized by spatially different routes.

1.3 Our contribution

In this paper, we introduce a novel algorithm to CF-ERP based on the idea of the Node-Independent Max-Flow optimization. Our approach has three main components: 1) Shortest Directed Acyclic Graph (S-DAG), 2) Node-Independent Max-Flow (NI-MF), and 3) Network-Cut based on multiple source and destination nodes. Specifically, our contributions are as follows:

- We introduce a new evacuation routing problem, namely Conflict-Free Evacuation Route Planning (CF-ERP).
- We propose a Node-Independent Max-Flow (NI-MF) algorithm for CF-ERP.
- We propose a novel Node-Weight based Node-Independent Route (NW-NIR) method to reduce the computational cost of NI-MF.

– We theoretically evaluate all proposed algorithms through cost models and proofs of algorithm properties (e.g., termination and correctness)
– We experimentally evaluate all proposed algorithms using five real-world road maps.

1.4 Related work

There has been a considerable amount of research work on evacuation route planning (ERP) problems in recent years. Recent work on evacuation route planning can be divided into three categories: (1) Network Flow Optimization methods that use network flow optimization techniques (e.g., max-flow or min-cost flow) to minimize the evacuation time, (2) Simulation methods that model the evacuation routes as individual movements or as a traffic assignment problem, and (3) Heuristic methods that use approximate optimization techniques to reduce the computational cost. Network Flow Optimization methods use a Spatiotemporal Network (STN) model and iteratively identify the optimal evacuation time using max-flow or min-cost flow optimization techniques [1, 4, 10, 11, 15, 17, 18]. The limitation of these methods is that the size of the STN increases as the optimal evacuation time increases. This limits its usefulness for large scale networks. Simulation methods use individual traveler behaviors or traffic assignments in greater detail including the interaction between vehicles [3, 22]. One problem with this model is that regulating individual movements or assigning traffic flow associated with Wardrop's equilibrium model in an emergency evacuation is very complicated, making it inappropriate for large evacuation scenarios [24, 29]. Besides, simulation studies for small areas without having a global perspective may not prevent potential risks of traffic congestions as we see in real-life. Lastly, heuristic methods can be used to incorporate capacity constraints into route planning and find near-optimal evacuation plans with reduced computational cost. This is useful for medium-size transportation networks within a limited amount of time. A well-known approach for this category is the Capacity Constrained Route Planner (CCRP) [19–21, 26, 31]. Unfortunately, all these approaches have no conflict-free constraint and simply ignore the topological interaction among evacuation routes, which leads to potential risks (e.g., traffic congestion, movement-conflicts, etc.) at road intersections and causes major delays.

The most important constraint of CF-ERP is to prevent potential movement-conflicts (e.g., merging or crisscrossing conflicts) in the evacuation route schedule. The safest path problem has been studied to minimize the potential risks for traveling [2]. Some research has been conducted on ERP with conflict-free constraints. Lane-based evacuation route planning was studied to minimize movement-conflicts based on the Linear Programming model [6]. The key idea is to construct a lane-based static network model and identify the shortest distance routes based on the minimum-cost objective and conflict constraints. However, since this approach does not consider temporal capacity constraints, it cannot provide the evacuation time for a dynamic network model. The spatial partitioning method was used to divide the evacuation zone to minimize the number of movement-conflicts between groups of pedestrian evacuees [33]. This approach groups evacuees based on the spatial partitioning and minimizes the interaction between groups. However, this special case is inappropriate for transportation network problems because the grouping of vehicles cannot reduce movement-conflicts. Static network conflict-free ERP was investigated to minimize both the evacuation time and the number of movement-conflicts in a static network [16]. The general idea is to construct a suffix tree for network flow and modify the tree to reduce the number of movement-conflicts. However, this idea does not consider temporally-detailed movement-conflicts. In contrast with these approaches, our approach enforces the conflict-

-free constraint in a spatiotemporal network (STN) and tries to minimize the evacuation time.

1.5 Scope and outline

This paper focuses on the spatiotemporal network optimization problem based on the conflict-free constraint. We model the transportation network using fixed travel times and maximum road capacities [9]. The capacity constraints and conflict-free constraint are restricted to a spatiotemporal network (STN). We consider that the transportation network exhibits the FIFO (First-In-First-Out) property. The simulation of individual behaviors and environmental dynamics is beyond the scope of the present research.

The rest of the paper is organized as follows: Section 2 describes our proposed approaches to CF-ERP. We provide correctness proofs of the proposed approaches in Section 3. Section 4 describes the experiment design and presents the experimental observations and results. Finally, Section 5 concludes the paper.

2 Proposed algorithms for CF-ERP

In this section, we introduce our Node-Independent Max-Flow (NI-MF) approach to the CF-ERP problem. Then we present a novel method that can reduce the computational cost of NI-MF.

2.1 Basic concept

Evacuation routes can be conceptualized as a flow with capacity constraints on a spatiotemporal network (STN). STN can be represented as a spatial graph with temporal attributes, composed of nodes (N), edges (E), and discrete time-points. Every node represents a spatial location in geographic space and every edge represents a connection between two nodes and has temporal attributes associated with it. We refer to an edge in STN as a temporal-edge. Similarly, we refer to a node in STN as a temporal-node.

Consider the spatial graph with the capacity constraints in Fig. 3a. The example spatial graph consists of 9 nodes and 12 edges. Every edge has a travel time and a capacity constraint. To simplify the example, we assume that every edge has a unit capacity. This simplification does not affect the generality of the STN model. Assume that node A is a source node and node I is a destination node. The spatial graph can be modeled as a time-expanded graph (TEG), that is, a spatiotemporal model that replicates each node along the time series such that a time-varying attribute is represented between replicated nodes (see Fig. 3b) [1, 10, 34]. For example, every temporal node in TEG is associated with a time-point (e.g., $A(0)$, $A(1)$, ..., $A(5)$). Every temporal-edge has a capacity and the flow along the edge cannot exceed the capacity. TEG is a directed acyclic graph (DAG) and neatly represents the temporally-detailed routes along time and space in STN. We can reduce the size of TEG using the time-aggregated graph storage model [14]. The shortest travel time route in STN can be defined as the route that reaches the destination at the earliest time-point. In the remainder of the paper, the shortest route is the shortest travel time route in STN.

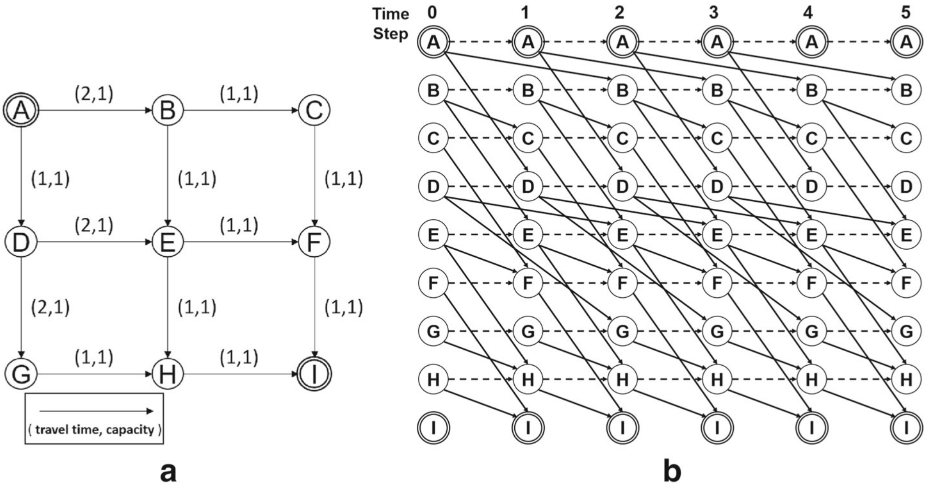


Fig. 3 STN as a time expanded graph

2.2 Node-Independent Max-Flow algorithm

The Node-Independent Max-Flow (NI-MF) algorithm starts with constructing a spatiotemporal network (STN) and pushes a flow along the STN that can minimize the evacuation time and honor the conflict-free constraint. NI-MF incrementally increases the starting time-point (or departure time-point) of evacuees and pushes flows to their destinations. This process requires multiple iterations to send all evacuees to their destinations.

The first core idea in NI-MF is to identify all possible node-independent shortest routes at each starting time-point and create a flow with no movement-conflicts. A set of routes from a source node s to a destination node d are node-independent if none of the routes share any nodes aside from s and d; therefore node-independent routes are equivalent to conflict-free routes [8]. NI-MF constructs a set of shortest routes that arrive at the same time and retrieves the maximum number of node-independent routes (see Lemma 1).

Consider again the input example in Fig. 3b. Assume that two evacuees try to move from node A to node I at a starting time-point of 0. Figure 4a shows an example of the shortest routes with no movement-conflicts (i.e., $A(0) \rightarrow B(2) \rightarrow C(3) \rightarrow F(4) \rightarrow I(5)$ and $A(0) \rightarrow D(1) \rightarrow G(3) \rightarrow H(4) \rightarrow I(5)$), whereas Fig. 4b shows two shortest routes that are crisscrossing at node $E(3)$.

NI-MF utilizes the special structure of STN to identify a set of node-independent shortest routes. It is important to note that a sub-graph of STN forms a directed acyclic graph (DAG) because STN is a DAG. First, NI-MF creates the shortest directed acyclic graph (S-DAG) based on the generalized Dijkstra's algorithm and retrieves all possible shortest routes. Figure 5a shows an example of all possible shortest routes and Fig. 5b shows an S-DAG consisting of 9 nodes and 12 edges.

NI-MF then splits a node whose in-degree is greater than one. In this example, nodes E, F, and H were each split into two nodes (i.e., (E^+, E^-), (F^+, F^-), and (H^+, H^-)) because all three nodes have an in-degree of 2 (see Fig. 6a). Next, NI-MF assigns unit capacity to all edges in the S-DAG and constructs a unit capacity directed acyclic graph.

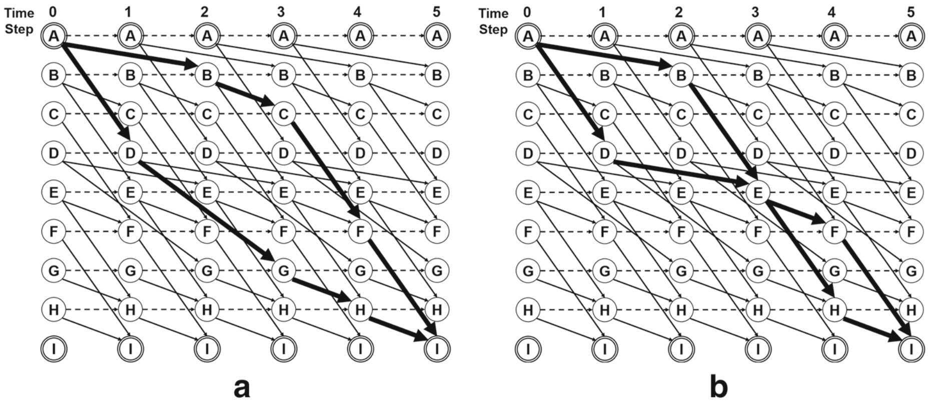


Fig. 4 Shortest Routes at a starting time-point of 0

This graph transformation allows the unit-capacity max-flow algorithm to identify all node-independent routes in the S-DAG (see Lemma 2). Figure 6b shows two node-independent routes based on the output of the unit-capacity max-flow computation. Since every node in S-DAG is uniquely mapped to a temporal-node in STN (see Lemma 3), we can transform these two routes into temporally-detailed shortest routes in the STN (i.e., $A(0) \rightarrow B(2) \rightarrow C(3) \rightarrow F(4) \rightarrow I(5)$ and $A(0) \rightarrow D(1) \rightarrow G(3) \rightarrow H(4) \rightarrow I(5)$) (see Fig. 4a). Note that these node-independent routes are not necessarily unique, so that there may be more than one way of choosing a set of node-independent routes.

The second core idea in NI-MF is the modification of the STN to enforce the conflict-free constraint for the next iteration. The STN modification has three steps: First, it reduces the capacity of the temporal-edges on the node-independent shortest routes based on the amount of flow. When the capacity of a temporal-edge becomes zero, NI-MF removes the temporal-edge from the STN. We refer to this edge as a saturated temporal-edge. This edge removal makes NI-MF honor the capacity constraint. Next, NI-MF identifies nodes whose in-degree is greater than one and removes all their incoming temporal-edges except the

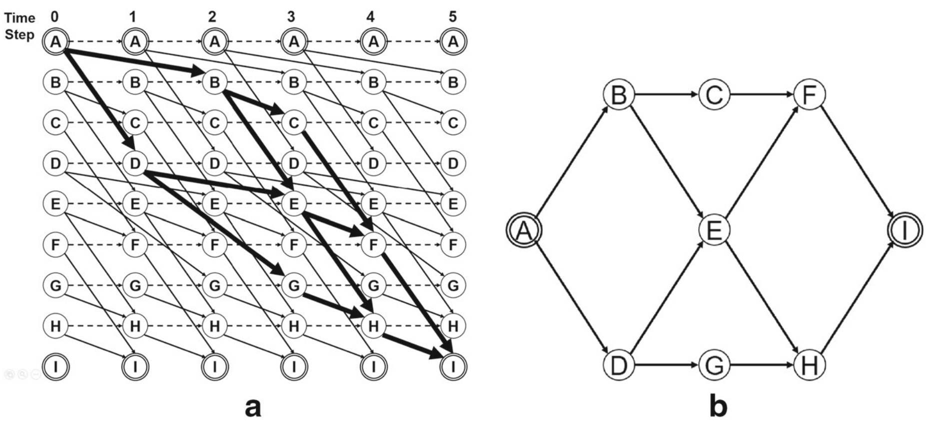


Fig. 5 Shortest Directed Acyclic Graph (S-DAG)

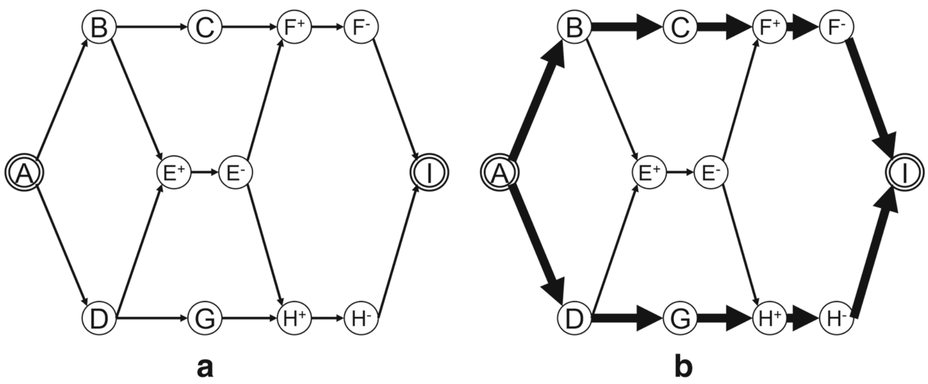


Fig. 6 Node-independent Routes computation

temporal-edges on the shortest route. This edge removal is important to ensure the node-independency among the shortest routes throughout all the iterations. Finally, it removes all temporal-nodes at the starting time-point of the shortest routes. This node removal reduces the size of the STN without losing information for the next iteration. The STN modification allows NI-MF to honor the conflict-free constraint at each iteration (see Lemma 1).

Consider the node-independent shortest routes in Fig. 4a. Figure 7a shows the output of the STN modification. In this example, we first remove 8 temporal-edges on the shortest routes (i.e., $\overrightarrow{A(0)B(2)}$, $\overrightarrow{B(2)C(3)}$, $\overrightarrow{C(3)F(4)}$, $\overrightarrow{F(4)I(5)}$, $\overrightarrow{A(0)D(1)}$, $\overrightarrow{D(1)G(3)}$, $\overrightarrow{G(3)H(4)}$, and $\overrightarrow{H(4)I(5)}$) because the capacity of these edges is saturated. We then remove the incoming edges of nodes $F(4)$ and $H(4)$ (i.e., $\overrightarrow{E(3)F(4)}$ and $\overrightarrow{E(3)H(4)}$). Lastly, we remove all temporal-nodes at a time-point of 0.

There are two key advantages behind the STN modification. First, it represents the STN space compactly and efficiently to reduce the storage cost (see Lemma 4). We can utilize the time-aggregated graph storage model and reduce the size of STN by limiting the minimum and maximum time-points [14]. Therefore, NI-MF does not use TEG for the node-indendent route computation; Instead it uses the time-aggregated graph with the limited time bound.

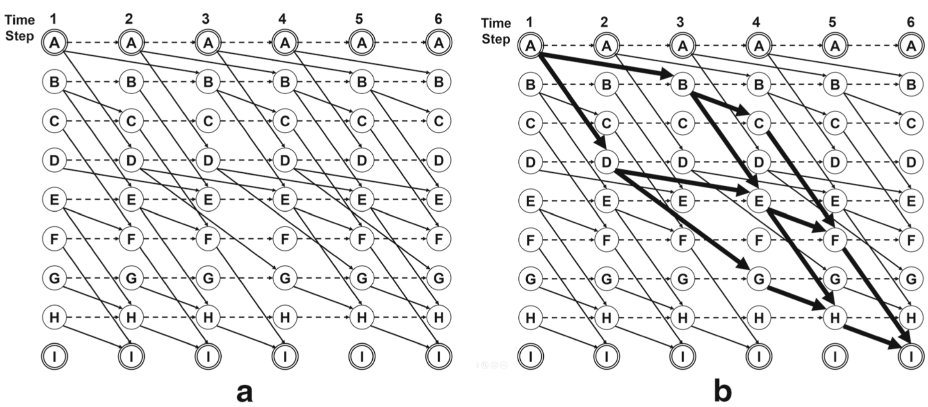


Fig. 7 Node-independent routes on DAG

The second key advantage is that NI-MF guarantees that all shortest routes are independent of each other through all time-points (see Lemma 1). Figure 7b shows all possible shortest routes at a starting time-point of 1 on the modified STN in Fig. 7a. These shortest routes are independent of the node-independent routes with a starting time-point of 0 in Fig. 4a.

The third core idea of NI-MF is to construct an S-DAG based on multiple sources and destinations to find as many node-independent shortest routes as possible at each iteration. Many road networks have low degree intersections, which makes it hard to retrieve many node-independent routes. It is known that the mean degree of intersections in the U.S. interstate road network is only about 2.86 [13]. Therefore we can retrieve 2.86 node-independent routes on average. One way to remedy the low degree issue is to construct an S-DAG using multiple source and destination nodes.

Consider the network model for the Evacuation Zone (EZ) in Fig. 8a. The source nodes inside of the EZ are $(G, H, I, L, M, N, Q, R, S)$ and the destination nodes are (A, E, U, Y). Figure 8b shows an example of S-DAG based on multiple sources and destinations. The outside nodes (G, I, Q, S) have shorter routes available compared to the inside nodes (H, L, N, R), reflecting an "outer first, inner last" flow pattern based on the network's FIFO (First-In-First-Out) property. That means, after evacuees from the boundary areas move out of their areas, there is a second wave of evacuees from inner regions into the boundary areas that will prepare to move out of the EZ. We can conceptualize this pattern using network-cuts. As can be seen in Fig. 8b, $Cut1$ increases the node degree between source and destination nodes, making it possible to find many node-independent shortest routes. In this example, 8 node-independent shortest routes are produced as shown by the thick arrow lines (see Fig. 8b).

The network-cut incrementally moves toward the inside of the EZ and finds another set of node-independent shortest routes. For example, Fig. 9a shows that $Cut2$ produces 8 routes for nodes (H, L, N, R) and Fig. 9b shows that $Cut3$ produces 4 routes for node M. These network-cuts enable us to identify multiple node-independent routes with a single execution of the unit-capacity max-flow computation. For example, we can group source nodes and destination nodes within the network-cut with super-source node and super-sink node respectively, and run the unit-capacity max-flow algorithm to find all node-independent routes.

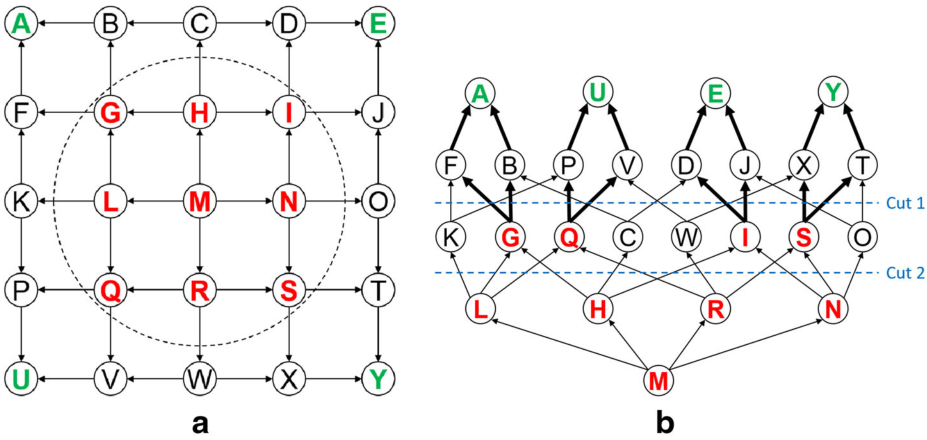


Fig. 8 Node-Independent Routes on S-DAG

Algorithm 1 Generalized Node-Independent Max-Flow algorithm (NI-MF) (pseudo-code).

```
Inputs:
  – A set of nodes N and edges E with capacity constraints C
  – Every edge e ∈ E has a travel time t
  – A set of source nodes S including initial evacuee occupancy O and
  – A set of destination nodes D
Outputs: Evacuation plan including route schedules for evacuees on each route r
Steps:
 1: while any source node s ∈ S has evacuees do
 2:    create S-DAG based on multiple source and destination nodes.
 3:    split nodes in S-DAG whose in-degree is greater than one and assign unit
       capacity to all edges.
 4:    find all node-independent shortest routes R in S-DAG.
 5:    for r ∈ R do
 6:       compute the minimum route capacity C_min along the route r.
 7:       identify feasible flow f = min( no. of remaining evacuees at s in r, C_min).
 8:       reduce the node and edge capacity c along the route r using f.
 9:       remove evacuees from O using f.
10:    end for
11:    modify the STN to enforce the conflict-free constraint for the next iteration.
12: end while
13: return the route schedules.
```

Algorithm 1 presents the pseudo-code for a generalized version of NI-MF. The input for the algorithm is a transportation network consisting of nodes, edges, source nodes, destination nodes, and capacity constraints. The output is a set of evacuation route schedules containing a sequence of nodes and edges with time-points. First, NI-MF creates an S-DAG based on multiple source and destination nodes (Line 2). It then splits nodes whose in-degree is greater than one in the S-DAG and assigns unit capacity to all edges (Line 3). Next, it finds all node-independent shortest routes R in the S-DAG using the unit-capacity max-flow computation (Line 4). After that, it computes the minimum route capacity for every route and reduces capacities along these routes as well as the evacuees of source nodes (Lines 6-9). Finally, it modifies the STN by removing temporal-nodes and temporal-edges for the

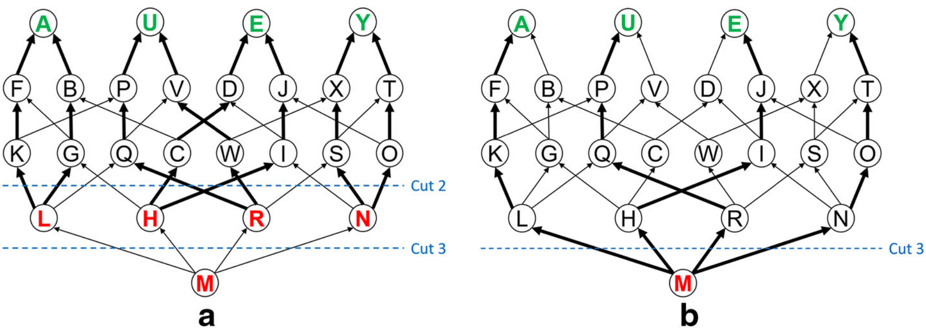


Fig. 9 Node-independent Routes on S-DAG

next iteration (Line 11). This process continues until we finish finding the routes for all remaining evacuees (Line 1).

2.3 Node-Weight Based Node-Independent Route computation

NI-MF utilizes the unit-capacity max-flow algorithm to identify a set of node-independent shortest routes. It splits the nodes in the S-DAG and assigns unit capacity to the S-DAG edges to create a unit capacity directed acyclic graph. However, this graph transformation is not practical when handling large-sized transportation networks with a large number of evacuees. In this subsection, we introduce a new method to reduce the computational cost of this step. The core idea of the proposed approach is to assign two node-weights to every node in the original S-DAG and iteratively retrieve the minimum weight routes to construct a set of node-independent routes. We refer to this as Node-Weight based Node-Independent Route computation (NW-NIR).

NW-NIR uses the idea that a node with high in-degree potentially causes movement-conflicts (e.g., merging or crisscrossing conflicts) and a node with high out-degree has many alternative routes to destinations. NW-NIR starts by assigning the two node-weights (i.e., in-degree and out-degree weights) to every node in the S-DAG. NW-NIR defines the in-degree weight of a node as the following function: $w_{in}(n) = \mathit{in\text{-}degree}(n) - 1$. This is because a node with an in-degree of 1 never causes movement-conflict. Similarly, it defines the out-degree weight of a node as the following function: $w_{out}(n) = \mathit{out\text{-}degree}(n) - 1$ because a node with an out-degree of 1 has a single choice to move to another location.

First, NW-NIR assigns two node-weights to every node except source and destination nodes and then iteratively identifies a single best candidate route for constructing a set of node-independent routes according to the bi-objective lexicographic method. That means, it identifies a set of the minimum in-degree weight routes and then selects the minimum out-degree weight route among them as a tie-breaker. It then removes all nodes on the chosen route and updates the two node-weights based on the change in the S-DAG. This process continues until it finds all node-independent routes. There are two key advantages of NW-NIR. First, we can utilize the directed acyclic property of S-DAG and compute the minimum-weight route using Dynamic Programming in linear time [5]. Second, NW-NIR is quite simple to implement compared to NI-MF because it does not require the graph transformation, including node-split and unit capacity assignment.

Consider the example of S-DAG in Fig. 5b. Figure 10a shows the node-weighted S-DAG. Since nodes E, F, and H have an in-degree of 2, we set their in-degree weight to 1. Similarly, since nodes B, D, and E have out-degree of 2, we set their out-degree weight to 1. First, given the weighted S-DAG, NW-NIR selects a single minimum weight route. In this example, we choose route $A \rightarrow B \rightarrow C \rightarrow F \rightarrow I$ as one candidate for node-independent routes (see Fig. 10b).

NW-NIR then removes all nodes on the minimum weight route and updates the weights of nodes in the S-DAG (see Fig. 11a). Next, NW-NIR computes the minimum weight route and produces a route that is independent of the ones produced in the previous iteration. Figure 11b shows one possible minimum weight route (i.e., $A \rightarrow D \rightarrow G \rightarrow H \rightarrow I$). This process continues until NW-NIR can identify all routes in the S-DAG. In this example, NW-NIR produces two node-independent shortest routes in two iterations.

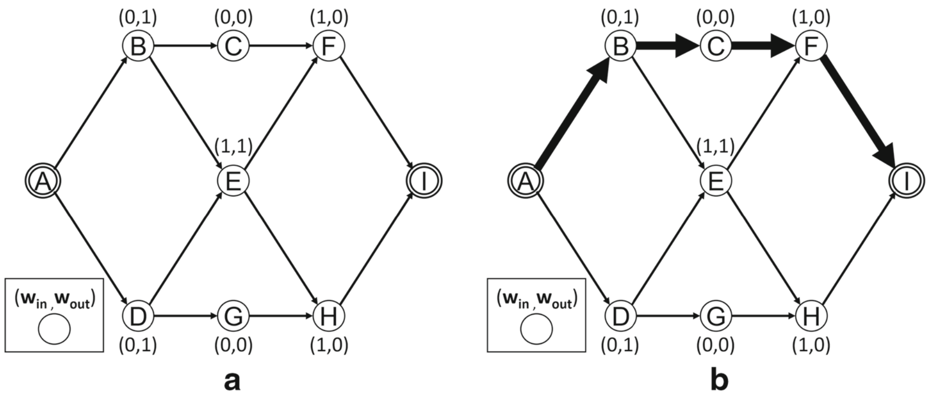


Fig. 10 Node-Independent Routes on S-DAG

3 Analysis on the quality of the proposed approaches

In this section, we prove the correctness of NI-MF and NW-NIR, and provide the algebraic cost model of the proposed approaches.

3.1 Proof and analysis

The following lemmas prove the correctness of NI-MF and NW-NIR approaches.

Lemma 1 *NI-MF creates conflict-free evacuation routes.*

Proof First, we argue that NI-MF creates a set of node-independent shortest routes at each iteration. Before moving to the next iteration, NI-MF removes the temporal-edges that may cause movement-conflicts with the routes produced by the next iteration. Therefore, NI-MF produces node-independent routes throughout all iterations. □

Lemma 2 *The unit-capacity max-flow algorithm produces the maximum node-independent routes in S-DAG by splitting nodes whose in-degree is greater than one and assigning unit capacity to all edges.*

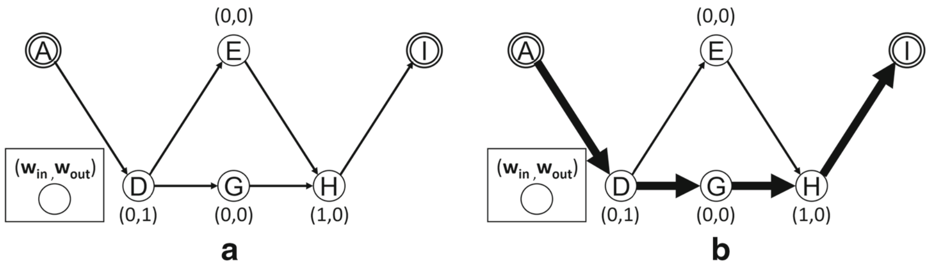


Fig. 11 Node-Independent Routes on S-DAG

Proof The unit-capacity max-flow algorithm produces the maximum edge-independent routes in a directed graph [1]. If two different routes share a node, then the value of in-degree of the shared node should be greater than one. Therefore, the maximum node-independent routes can be obtained by splitting nodes whose in-degree is greater than one and assigning unit capacity to all edges because the node-split makes it obvious that the node exists uniquely among a set of routes. □

Lemma 3 *Every node in S-DAG is uniquely mapped to a temporal-node in STN.*

Proof The shortest routes between two nodes (i.e., source and destination nodes) in STN is unique. NI-MF forms an S-DAG based on these shortest routes. Assume that the S-DAG includes two duplicate nodes. Since the time-point of these two duplicate nodes is not the same, NI-MF produces a longer or shorter route than the shortest route. This contradicts that the S-DAG is constructed based on a set of shortest routes. Therefore, we claim that every node in S-DAG is unique and uniquely mapped to a temporal-node in STN. □

Lemma 4 *The storage cost of STN in NI-MF is $O(n \cdot st(G))$, where n is the number of nodes and $st(G)$ is the shortest travel time in STN.*

Proof NI-MF incrementally expands a STN using the shortest path computation and finds a set of shortest routes. At each iteration, NI-MF stops the expansion when it reaches destination nodes with the shortest travel time. Let $st(G)$ be the shortest travel time in the STN. Then the size of the STN for constructing an S-DAG is bounded by $O(n \cdot st(G))$ because the number of time-points is bounded by $st(G)$. □

Lemma 5 *NI-MF terminates and produces conflict-free evacuation routes.*

Proof At each iteration, NI-MF identifies at least one shortest route and sends at least one evacuee along this route. Since the number of evacuees is bounded by p, NI-MF terminates after at most p iterations and produces conflict-free evacuation routes according to Lemma 1. □

Lemma 6 *NW-NIR produces results similar to those of NI-MF.*

Proof NW-NIR uses a heuristic method to identify a set of node-independent shortest routes. Since NW-NIR creates a set of routes that meet the conflict-free constraint while minimizing the evacuation time for the population, it produces results similar to those of NI-MF. □

3.2 Asymptotic analysis of the proposed algorithms

We developed a cost model for the proposed approaches to estimate their computational cost.

3.2.1 NI-MF

Let n be the number of nodes, let m be the number of edges, and let p be the number of evacuees. First, NI-MF constructs an S-DAG using the generalized Dijkstra's algorithm, which takes $O(n \cdot \log n + m)$. It then modifies the S-DAG and retrieves the node-independent routes from the S-DAG using the unit-capacity max-flow computation [1]. This takes $O(p \cdot (n + m))$ for all iterations. Finally, NI-MF reduces the capacities of temporal-edges on the routes and modifies the STN to remove unnecessary temporal-nodes and edges for the next iteration, which takes $O(n)$. Since the number of iterations is bounded by p, the cost model of NI-MF is $O(p \cdot (n \cdot \log n + m))$.

3.2.2 NW-NIR

The main difference between NW-NIR and NI-MF is the node-independent route computation. First, NW-NIR constructs an S-DAG and assigns two node-weights to every node except source and destination nodes in the S-DAG. This step takes $O(n)$. It then identifies node-independent routes using the minimum-weight route computation. The minimum-weight route computation takes $O(m)$ in the S-DAG based on Dynamic Programming [5]. Since the number of node-independent routes is bounded by p, the total cost for the node-independent route computation is bounded by $O(p \cdot m)$. Therefore, the cost model of NW-NIR is $O(p \cdot (n \cdot \log n + m))$. The cost model of NW-NIR is the same as that of NI-MF. However, NW-NIR does not require the graph transformation, which can reduce the computation time for creating a set of node-independent routes.

4 Experimental evaluation

We conducted experiments to evaluate the performance of the NI-MF and NW-NIR approaches. We used the Capacity Constrained Route Planning (CCRP) algorithm as a baseline for measuring the performance of the proposed approaches [20, 21]. As we reduce the number of merging or crisscrossing points in network flow, the network clearance time (i.e., evacuation time) increases. However, these conflict points cause potential risks, including traffic accidents, congestion, and delays. It is important to note that CCRP has no conflict-free constraint; therefore it can serve as a lower-bound of the optimal evacuation time of CF-ERP.

We wanted to answer five questions: (1) What is the effect of the number of evacuees? (2) What is the effect of the number of source nodes? (3) What is the effect of the size of the network? (4) What is the effect of the number of destination nodes? and (5) Is the NI-MF algorithm correct, and is the solution quality preserved?

4.1 Experiment layout

Figure 12 shows the experimental setup for the CF-ERP problem. We chose five different municipal areas in the U.S. from OpenStreetMap [30]. We constructed multiple sub-networks from each transportation network to vary the network size. Table 1 shows the number of nodes (i.e., $|N|$) and edges (i.e., $|E|$) in the chosen areas. We extracted information about road properties (e.g., road type, the number of lanes, etc.) and constructed

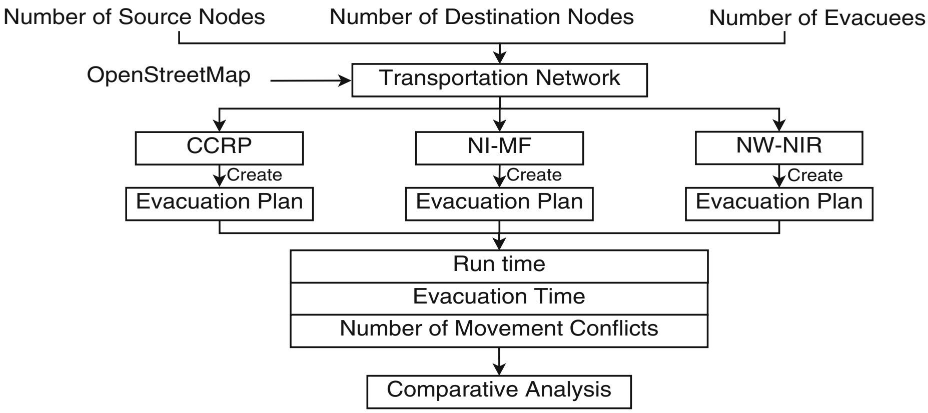


Fig. 12 Experimental layout

travel times and capacity constraints for the road networks. In our experimental analysis, we fixed the travel times and capacity constraints according to road properties. The uncertainty aspects of travel times during evacuation are beyond the scope of the present research [23]. We tested three approaches: (a) Capacity Constrained Route Planner (CCRP), (2) Node Independent-Maximum Flow (NI-MF), and (3) Node Weight based Node Independent Route (NW-NIR). The algorithms were implemented in Java 1.8 with an 8 GB memory run-time environment. All experiments were performed on an Intel(R) Xeon(R) W-2175 CPU @ 2.50GHz machine running Windows 10 with 64 GB of RAM.

4.2 Experiment results and analysis

We experimentally evaluated the proposed algorithms by comparing the impact on the performance of (1) the number of evacuees, (2) the number of source nodes, (3) the size of the network, and (4) the number of destinations.

4.2.1 Effect of the number of evacuees

The first experiment evaluated the effect of the number of evacuees on the performance of the algorithms. Performance measurements were execution time, evacuation time, and the number of movement-conflicts. We used the five road maps listed in Table 1. First, we fixed the number of source nodes to 3,000 and the number of destination nodes to 60. We then incrementally increased the number of evacuees at every source node. The number of

Table 1 Transportation networks datasets (source: OpenStreetMap [30])

Area	No. of nodes ($\|N\|$)	No. of edges ($\|E\|$)
Fort Lauderdale, FL	30,668	84,598
Miami, FL	40,484	114,822
Cape Cod, MA	42,104	107,566
Boston, MA	44,298	125,040
New Orleans, LA	55,107	166,184

evacuees was varied from 30,000 to 150,000. The execution times were averaged over 10 test runs.

Figure 13a gives the run-times. As the number of evacuees increases, the run-time for the three approaches increases. As can be seen, both NI-MF and NW-NIR outperform CCRP. This is because these approaches identify multiple node-independent shortest routes at each iteration, whereas CCRP identifies a single shortest route. NW-NIR performs better than NI-MF because it can reduce the computational cost for finding node-independent routes in S-DAG. Figure 13b shows the evacuation time and Fig. 13c shows the number of movement-conflicts. Figure 13b shows that CCRP has a lower evacuation time compared to NI-MF and NW-NIR. The output of CCRP can serve as a lower bound for CF-ERP because CCRP does not honor the conflict-free constraint. It is important to note that CCRP produces a large number of movement-conflicts, whereas both NI-MF and NW-NIR have no movement-conflicts (see Fig. 13c). Essentially, these movement-conflicts cause delays at intersections during evacuation and result in a longer evacuation time. As the number of evacuees increases, both the run-time and the number of movement-conflicts for CCRP

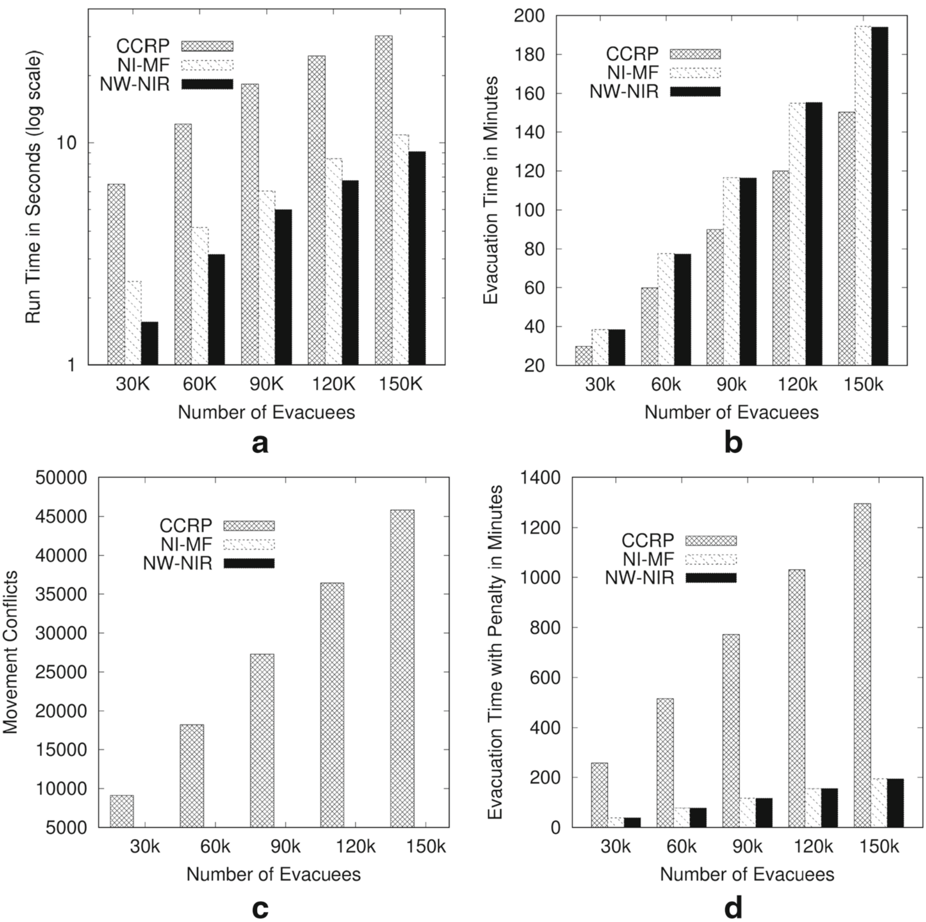


Fig. 13 Effect of No. of Evacuees ($|S| = 3,000$ and $|D| = 60$)

increase. To understand the effect of the conflict-point delay, we assigned a 1.5 s vehicle delay to every conflict point [12]. Figure 13d shows the evacuation time with conflict delays. As can be seen, both NI-MF and NW-NIR significantly outperform CCRP in terms of the evacuation time.

4.2.2 Effect of the number of source nodes

The second set of experiments evaluated the effect of the number of source nodes on algorithm performance. Performance measurements were execution time, evacuation time, and the number of movement-conflicts. We used the five road maps listed in Table 1. First, we fixed the number of destination nodes to 60 and the number of evacuees to 60,000. We then varied the number of source nodes from 1,000 to 4,000. The execution times were averaged over 10 test runs.

Figure 14a shows the run-times of the three approaches. As the number of source nodes increases, the run-time of the three approaches also increases. We can see that both NI-MF and NW-NIR significantly outperform CCRP. Since NI-MF and NW-NIR utilize the network cut to increase the degree between source and destination nodes, both approaches can compute many node-independent shortest routes at each iteration and reduce the number of iterations for identifying node-independent shortest routes. NW-NIR performs better than NI-MF due to the lower computational effort for finding node-independent routes. Figure 14b and c show the solution quality of the three approaches. Figure 14b shows that as the number of source nodes increases, the evacuation time decreases. This is because the number of evacuees at each source-node decreases as we increase the number of source-nodes. CCRP has a lower evacuation time compared to NI-MF and NW-NIR because CCRP does not honor the conflict-free constraint. However, Fig. 14c shows that CCRP produces a large number of movement-conflicts, whereas NI-MF and NW-NIR have no movement-conflicts. This large number of conflicts will increase the delay at intersections and increase the actual evacuation time. The number of movement-conflicts increases with the increase in the number of source nodes. Figure 14d shows the evacuation time with a 1.5 s conflict delay [12]. We can see that both NI-MF and NW-NIR significantly outperform CCRP.

4.2.3 Effect of the network size

The third set of experiments evaluated the effect of the size of the network. Performance measurements were execution time, evacuation time, and the number of movement-conflicts. We used the five road maps listed in Table 1. We incrementally increase the number of evacuees with the number of source nodes to show a proportional increase of the network. First, we fixed the number of destination nodes to 60 and the number of evacuees per a source node to 30. We then varied the number of source nodes from 1,000 to 4,000 and increased the number of evacuees from 30,000 to 150,000 proportional to the number of source nodes. Execution times were averaged over 10 test runs.

Figure 15a shows the run-times of the three approaches. As the size of the network increases, the run-time for the three approaches increases. As can be seen, both NI-MF and NW-NIR significantly outperform CCRP. Figure 15b shows that CCRP produces a lower evacuation time compared to our proposed approaches. However, CCRP produces a large number of movement-conflicts (see Fig. 15c) and will increase the actual evacuation time due to turning movement delays and potential crashes. The number of movement-conflicts

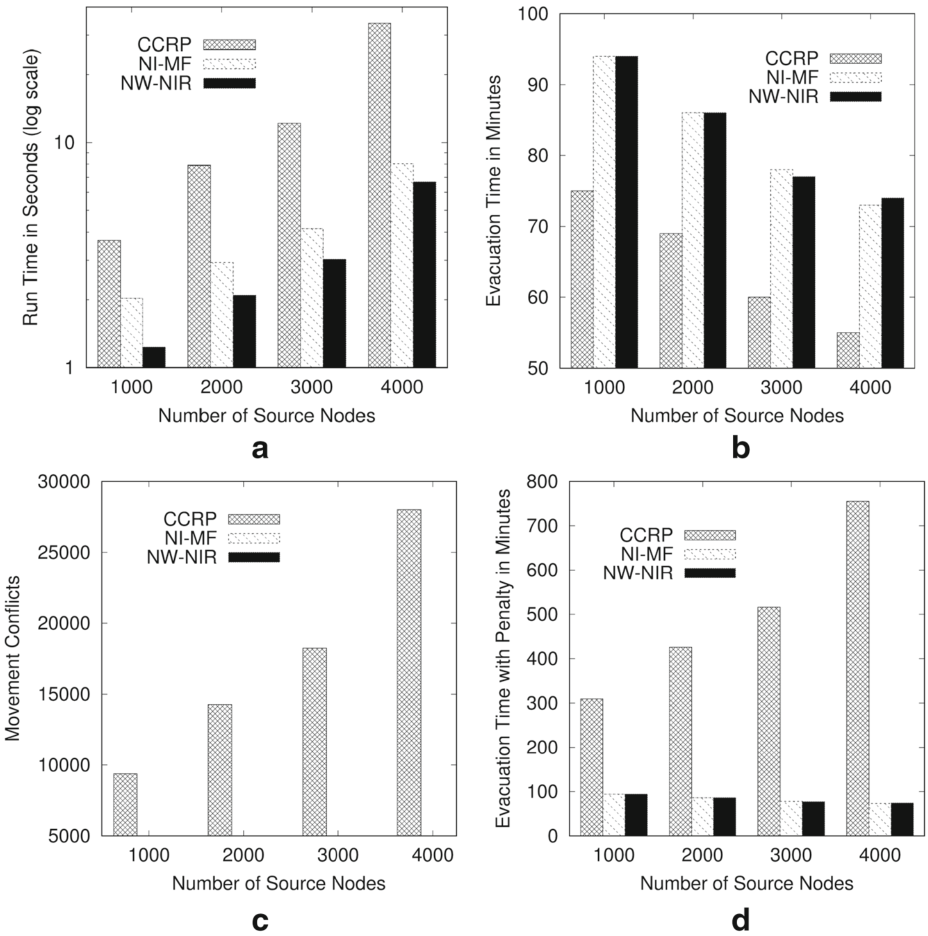


Fig. 14 Effect of no. of source nodes ($|D| = 60$ and $p = 60,000$)

for CCRP increases with the increase in the network size. NI-MF and NW-NIR again produce routes with no movement-conflicts. Figure 15d shows the evacuation time with a 1.5 s conflict delay [12]. It was shown that both NI-MF and NW-NIR perform better than CCRP in terms of the evacuation time if we take into consideration the delays caused by movement conflicts.

4.2.4 Effect of the number of destinations

The fourth set of experiments evaluated the effect of the number of destination nodes on algorithm performance. Performance measurements were execution time, evacuation time, and the number of movement-conflicts. We used the five road maps listed in Table 1. First, we fixed he number of source nodes to 3,000 and the number of evacuees to 60,000. We then varied the number of destination nodes from 20 to 100.

Figure 16a shows the run-times of CCRP, NI-MF, and NW-NIR. As the number of destinations increases, the run-time of the algorithms slightly decreases. This is because all

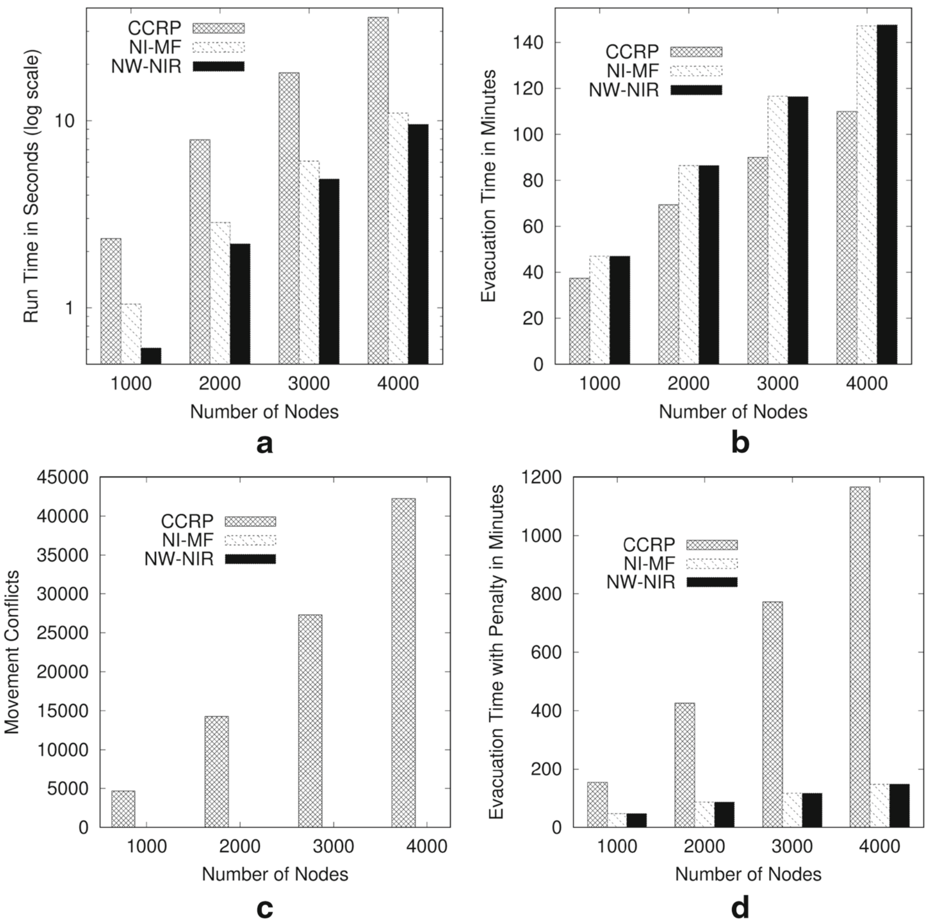


Fig. 15 Effect of network size ($|D| = 60$)

three algorithm can find more shortest routes as the number of destinations increases. We can see that both NI-MF and NW-NIR outperform CCRP. NW-NIR shows better performance than NI-MF due to the reduced computational effort for the node-independent route computation. Figure 16b shows that CCRP produces a lower evacuation time than NI-MF and NW-NIR because it does not honor the conflict-free constraint. As the number of destinations increases, the evacuation time decreases. Figure 16c shows that CCRP can reduce movement-conflicts with the increase in the number of destination nodes. This is because CCRP has a higher chance to find node-independent routes as the number of destination nodes increases. However, the number for movement-conflicts is still too high to reduce the potential risks (e.g., intersection delay and crashes). Figure 16d shows the evacuation time with a 1.5 s conflict delay [12]. The result of experiments shows that both NI-MF and NW-NIR significantly outperform CCRP.

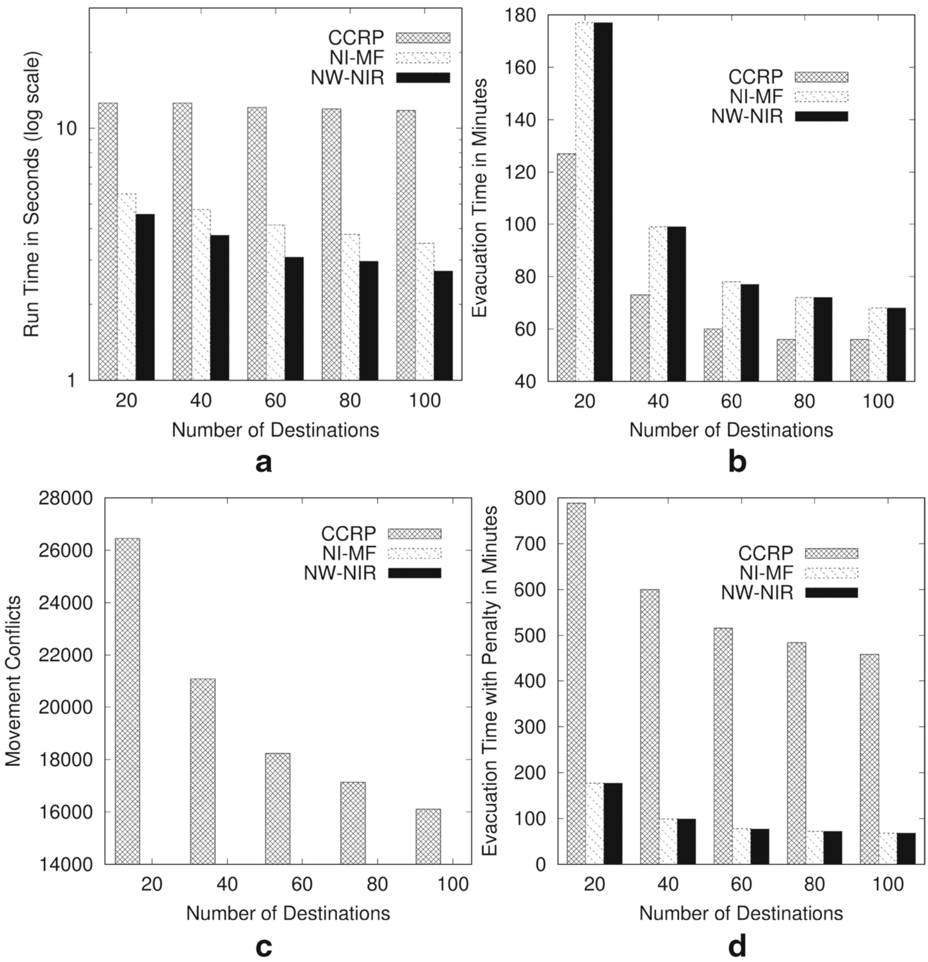


Fig. 16 Effect of no. of destinations ($|S| = 3,000$ and $p = 60,000$)

4.3 Discussion

The proposed NI-MF approach achieves a significant computation performance gain over the related work (i.e., CCRP) and produces evacuation routes with no movement-conflicts. This improvement has been achieved using three main components: 1) Shortest Directed Acyclic Graph (S-DAG), 2) Node-Independent Max-Flow (NI-MF), and 3) Network-Cut based on multiple source and destination nodes. NI-MF constructs an S-DAG and applies the unit-capacity max-flow algorithm to S-DAG to find as many node-independent shortest routes as possible at each iteration. It also applies the STN modification to honor the conflict-free constraint throughout all iterations. The experiments show that these novel components can significantly reduce the computational cost for the CF-ERP problem and produce routes with no movement-conflicts. We also proposed NW-NIR to reduce the computational cost for NI-MF. NW-NIR is a much simpler way to identify a set of node-independent routes in S-DAG. Our experimental analysis shows that NW-NIR performs better than NI-MF in terms of run-time.

5 Conclusion and future work

Evacuation route computation is a critical component for identifying the safest and most efficient routes in the wake of man-made or natural disasters (e.g., terrorist acts, hurricanes, or nuclear accidents). The key objective is to minimize the evacuation time for all expected residents in the evacuation zone. Previous work has focused on minimizing the evacuation time based on network flow optimization techniques. However, the output cannot reduce the potential movement-conflicts that cause traffic accidents, congestion, and delays. In this paper, we presented the Conflict-Free Evacuation Route Planning (CF-ERP) problem that minimizes the evacuation time with no movement-conflicts. The CF-ERP problem is computationally challenging because of the large size of the transportation network, the large number of evacuees, the capacity constraints of the network, and the conflict-free constraint on routes. We introduced the Node-Independent Max-Flow (NI-MF) approach to CF-ERP for producing evacuation routes to meet the conflict-free constraint while minimizing the evacuation time. We also proposed a novel Node-Weight based Node-Independent Route computation (NW-NIR) to reduce the computational cost of the NI-MF approach. Experiments using five different transportation networks demonstrated that our proposed algorithms outperform the related work.

In future work, we would like to explore a parallel formulation of NI-MF and utilize big data processing platforms to improve the scalability for continental sized networks. We plan to identify independent components of the node-independent route computation and develop a parallel algorithm for the CF-ERP problem. We will also investigate the uncertainty of evacuation time caused by traffic load, weather condition, or traffic accident. The recent growth of trajectory data enables us to incorporate the travel time uncertainty into the transportation network model [7, 32]. Since the uncertainty of travel time changes the topological structure of the time-expanded graph (TEG), the flow optimization methods should consider new spatiotemporal network models and optimization techniques. Lastly, we plan to study a model that can identify potential traffic bottlenecks and identify the most vulnerable areas during emergency evacuations.

Acknowledgements We would like to thank the National Science Foundation CAREER under Grant No. 1844565. We also thank Prof. Shashi Shekhar from University of Minnesota for his helpful comments and discussions.

References

1. Ahuja R, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms and applications. Prentice-Hall, New Jersey
2. Aljubayrin S, Qi J, Jensen CS, Zhang R, He Z, Wen Z (2015) The safest path via safe zones. In: 2015 IEEE 31st international conference on data engineering. IEEE, pp 531–542
3. Ben-Akiva M et al (2002) Development of a deployable real-time dynamic traffic assignment system: dynamit and dynamit-p user's guide. Intelligent Transportation Systems Program Massachusetts Institute of Technology

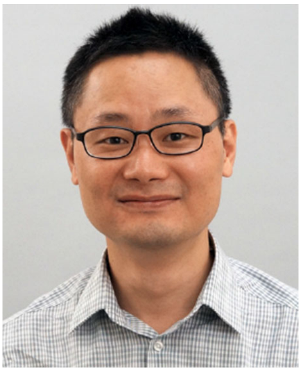
4. Burkard RE, Dlaska K, Klinz B (1993) The quickest flow problem. Z Oper Res 37(1):31–58
5. Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, MIT Press, Cambridge
6. Cova TJ, Johnson JP (2003) A network flow model for lane-based evacuation routing. Transp Res Part A: Policy Pract 37(7):579–604
7. Dai J, Yang B, Guo C, Jensen CS, Hu J (2016) Path cost distribution estimation using trajectory data. Proc VLDB Endow 10(3):85–96
8. Diestel R (2005) Graph theory. Grad. Texts in Math **101**
9. Fleischer L, Skutella M (2007) Quickest flows over time. SIAM J Comput 36(6):1600–1630
10. Ford LR Jr, Fulkerson DR (1958) Constructing maximal dynamic flows from static flows. Oper Res 6(3):419–433
11. Francis R, Saunders P (1979) Evacnet: prototype network optimization models for building evacuation (rep. no. nbsir 79-1593). Natl Bur Stand (US)
12. Furth PG, Cesme B, Muller THJ (2009) Lost time and cycle length for actuated traffic signal. Transp Res Rec 2128(1):152–160. https://doi.org/10.3141/2128-16
13. Gastner MT, Newman ME (2006) The spatial structure of networks. Eur Physi J B-Condens Matter Complex Syst 49(2):247–252
14. George B, Shekhar S (2008) Time-aggregated graphs for modeling spatio-temporal networks. In: Journal on Data Semantics XI. Springer, pp 191–212
15. Hamacher H, Tjandra S (2002) Mathematical modelling of evacuation problems: state of the art. In: Pedestrian and evacuation dynamics. Springer, pp 227–266
16. Herschelman R, Yang K (2019) Conflict-free evacuation route planner. In: Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 480–483
17. Hoppe B, Tardos É (1994) Polynomial time algorithms for some evacuation problems. In: SODA, vol 94, pp 433–441
18. Hoppe B, Tardos E (2000) The quickest transshipment problem. Math Oper Res 25(1):36–62
19. Kim S, George B, Shekhar S (2007) Evacuation route planning: scalable heuristics. In: Proceedings of the 15th annual ACM international symposium on advances in geographic information systems, pp 1–8
20. Lu Q, George B, Shekhar S (2005) Capacity constrained routing algorithms for evacuation planning: a summary of results. In: International symposium on spatial and temporal databases. Springer, pp 291–307
21. Lu Q, George B, Shekhar S (2007) Evacuation route planning: a case study in semantic computing. Int J Semant Comput 1(2):249–303
22. Mahmassani H, Sbayti H, Zhou X (2004) Dynasmart-p: intelligent transportation network planning tool: version 1.0 user's guide. Maryland Transportation Initiative, University of Maryland, College Park
23. Pedersen SA, Yang B, Jensen CS (2020) Fast stochastic routing under time-varying uncertainty. VLDB J 29(4):819–839
24. Pel AJ, Bliemer MC, Hoogendoorn SP (2012) A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. Transportation 39(1):97–123
25. Regan A (2017) Hurricane Irma could create one of the largest mass evacuations in U.S. history. https://www.usatoday.com/story/news/nation-now/2017/09/07/hurricane-irma-evacuations-florida/643045001/ Online; Accessed: 12 June 2020
26. Shekhar S, Yang K, Gunturi VM, Manikonda L, Oliver D, Zhou X, George B, Kim S, Wolff JM, Lu Q (2012) Experiences with evacuation route planning algorithms. Int J Geogr Inf Sci 26(12):2253–2265
27. Southworth F (1991) Regional evacuation modeling: A state-of-the art review (Technical Report). Center for Transportation Analysis Oak Ridge National Laboratory, Oak Ridge, TN, https://www.survivalring.org/classics/RegionalEvactuationModeling-StateOfTheArtReview-Mar1991-ORNL-TM-11740.pdf. Accessed 31 May 2020.
28. State of Florida (2018) Select Committee on Hurricane Response and Preparedness (Final Report). https://www.myfloridahouse.gov/. Online; Accessed 31 May 2020
29. Wardrop J (1952) Some theoretical aspects of road traffic research. In: Proceedings of the institution of civil engineers, 2(1). Thomas Telford Ltd
30. www.openstreetmap.org: Online; Accessed 31 Mar 2020
31. Yang K, Gunturi VM, Shekhar S (2012) A dartboard network cut based approach to evacuation route planning: a summary of results. In: International conference on geographic information science. Springer, pp 325–339
32. Yang B, Guo C, Jensen CS, Kaul M, Shang S (2014) Stochastic skyline route planning under time-varying uncertainty. In: 2014 IEEE 30th International conference on data engineering. IEEE, pp 136–147

33. Yang K, Shekhar AH, Rehman FU, Lahza H, Basalamah S, Shekhar S, Ahmed I, Ghafoor A (2015) Intelligent shelter allotment for emergency evacuation planning: a case study of makkah. IEEE Intell Syst 30(5):66–76
34. Yang K, Shekhar S (2017) Spatial network big databases: queries and storage methods. Springer, Cham

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Roxana Herschelman received a bachelor's and master's degree in computer science from Florida Atlantic University, FL, USA in 2016 and 2017, respectively. She received a Ph.D. degree in computer science from Florida Atlantic University in 2020. Her research interests include spatial big data analysis, spatial networks, spatial and spatiotemporal data mining, and routing algorithms.

Ahmad Qutbuddin received his BS degree in Computer Engineering from King Fahd University of Petroleum and Minerals in 2010. He worked as a Lecturer for Umm Al-Qura University in Makkah, Saudi Arabia from 2010 to 2014. He received a master's degree in Computer Engineering from the University of Central Florida in 2017. He is currently working toward his PhD degree in Computer Engineering at the Florida Atlantic University. His research interests are spatial network databases and Network Voronoi Diagram.



KwangSoo Yang is an assistant professor in the Department of Computer and Electrical Engineering and Computer Science at Florida Atlantic University. He received the BS degree in electrical engineering from Yonsei University in 1998, the master's degree in computer science from the University of Minnesota in 2010, and the PhD degree in computer science from the University of Minnesota in 2015. He was a software engineer for LG CNS in Seoul, Korea, from 2001 to 2008. He has received an NSF CAREER Award in 2019. His research interests include spatio-temporal network databases, spatio-temporal networks, and evacuation routing problems.