Intelligent Shelter Allotment for Emergency Evacuation Planning: A Case Study of Makkah

KwangSoo Yang, *Florida Atlantic University*

Apurv Hirsh Shekhar, *Johns Hopkins University*

Faizan Ur Rehman, *Umm Al-Qura University and University of Grenoble Alpes*

Hatim Lahza, *Umm Al-Qura University*

Saleh Basalamah, *Umm Al-Qura University*

Shashi Shekhar, *University of Minnesota*

Imtiaz Ahmed, *Umm Al-Qura University*

Arif Ghafoor, *Purdue University*

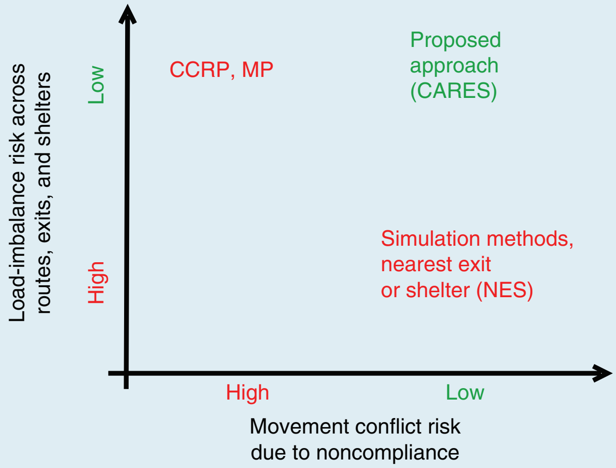
Intelligent shelter allotment faces challenges related to movement conflicts and transportation network choke points. A novel approach based on the idea of spatial anomaly avoidance provides faster evacuation.

Given maps of a vulnerable evacuee population, shelter locations, and a transportation network, the goal of intelligent shelter allotment (ISA) is to assign route and destination information to evacuee groups to minimize their evacuation time in the face of spatial disjointedness, the nonoverlapping separation of evacuation zones that's preferred by emergency managers to ensure smooth crowd movement. ISA can help in emergency planning and response by allocating shelters, exits, and routes. The goal is to speed up evacuation while reducing risks related to movement conflicts such as evacuation slowdowns, compression, and stampedes.

ISA faces numerous challenges, including bottlenecks and choke points in transportation networks (see Figure 1a), movement conflicts (when evacuee groups go to different exits or shelters), and scalability in terms of the number of evacuees and overall transportation network size. The current state of the practice is based on tabletop

1541-1672/15/$31.00 © 2015 IEEE
Published by the IEEE Computer Society

(a)



(b)

(c)

Figure 1. Intelligent shelter allotment (ISA). The nearest shelter could violate capacity constraints: (a) Newark Airport evacuation (www.doobybrain.com), (b) goals of our approach versus related work, and (c) pilgrims at Jamarat complex in the city of Makkah, Saudi Arabia (www.saudiembassy.ne).

exercises that help emergency managers manually identify and compare alternative routes, exits, and shelters. This traditional approach often leads to allotments based on a nearest exit or shelter (NES) paradigm[1] to minimize movement conflicts. However, to manually identify all choke points and bottlenecks in a transportation network is a daunting task. Thus, NES risks load imbalance, leading to unnecessarily high evacuation times, as Figure 1b shows.

We can categorize computational approaches for evacuation route planning into microscopic simulation,[2] mathematical programming (MP),[3,4] and capacity-constrained route planning (CCRP).[5] Microscopic simulation methods use agent-based models to capture human behavior under the assumption of perfect information and repeated experience to derive the game-theoretic Wardrop equilibrium, where no user may lower her travel time by unilateral action. However, such approaches are computationally exorbitant even for medium-sized evacuation scenarios with tens of thousands of evacuees. In addition, the assumption of repeated experience to achieve Wardrop equilibrium and perfect information doesn't hold for rare events such as emergency evacuations.

To reduce computational costs, MP methods simplify models by examining macroscopic traffic-flow behavior along transportation networks. Specifically, they define a mathematical program (such as a linear, integer, or quadratic program) on a time-expanded graph to replicate the transportation-flow-network model for every time unit during an evacuation.

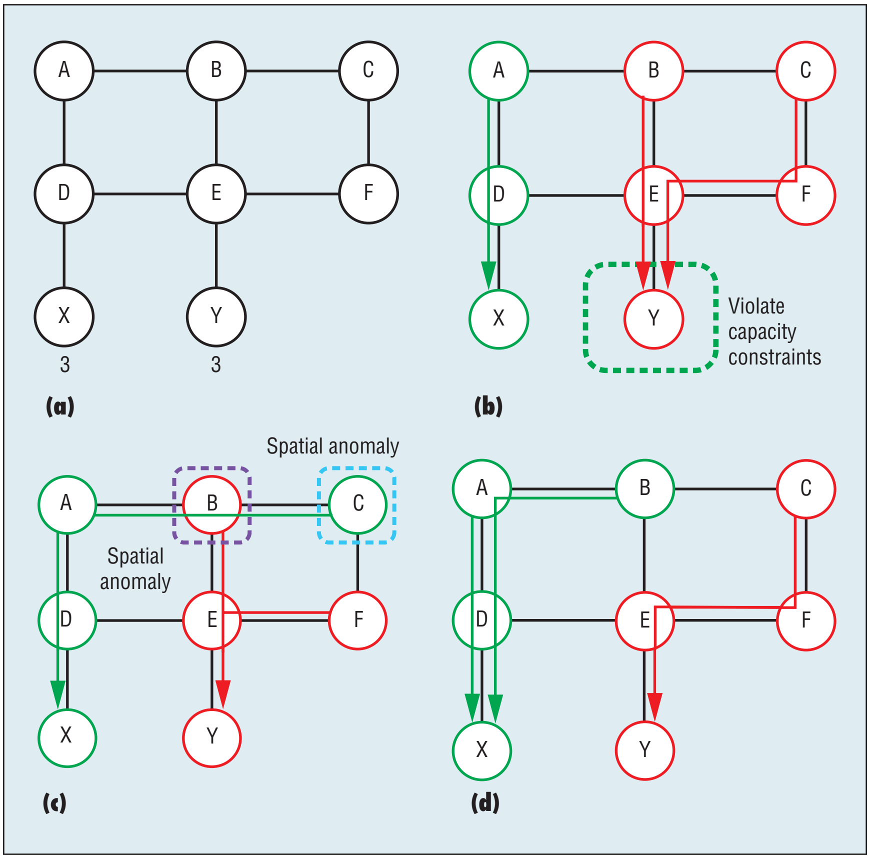


Figure 2. ISA input and possible outputs: (a) input, (b) allotment 1 (invalid), (c) allotment 2, and (d) allotment 3. Each shelter has capacity for three evacuees. For simplicity, edge travel time is one unit of time and capacity is one person per unit time.

Unfortunately, the computational cost for large evacuation scenarios with hundreds of thousands (or even millions) of evacuees can be prohibitive. In addition, traditional MP approaches require the user to set a upper bound on evacuation time in order to derive a solution, which might be difficult for untrained end users. CCRP addresses some of these limitations by using time-aggregated graphs that dramatically reduce the storage needs of time-expanded graphs. However, it uses spatially overlapping (although temporally disjointed) paths across different shelters, risking movement conflicts due to noncompliance by evacuees on their prescribed movement schedules (Figure 1b).

This article proposes a novel approach—Crowd-separated Allotment of Routes, Exits, and Shelters (CARES)—that's based on the core ideas of detection and avoidance of two kinds of spatial anomalies. Type I spatial anomalies are evacuation units (groups of evacuees) with an allotted route, exit, or shelter that differs from the ones allotted to all of its spatial neighbors. Type II spatial anomalies refer to violations of shelter service area contiguity, which implies that each evacuee route to designated service center c is contained in the service area of c. Because spatial anomalies are necessary conditions for movement conflicts, CARES identifies them and reconfigures routes, exits, and shelters when necessary. We performed an experimental evaluation and a case study regarding Hajj in the city of Makkah (Saudi Arabia) to show that CARES can significantly reduce the number of spatial anomalies relative to CCRP and give much faster evacuation times than typical NES scenarios. Although a real-time application is beyond the scope of the present study, in the future, CARES could be used to preplan routes and shelters under various evacuation scenarios.

The Problem Statement

We can formally define the ISA problem as follows.

Given

- an undirected graph $G = (V, E)$, where V is a set of nodes and E is a set of edges,
- edge travel time t: $E \rightarrow Z^+$,
- edge capacity/unit time, ec: $E \rightarrow Z^+$,
- shelters $x \in S$ subset of V, and
- shelter capacity, sc: $S \rightarrow Z^+$,

find allotment A: $V \rightarrow S$. The objective is to minimize the maximum travel time for any evacuee to reach his or her allotted shelter.

Constraints

For the ISA problem, the following statements are true.

- Regarding shelter service area contiguity, if $A(u) = x$, then there's a path $p(u, x)$ such that $A(a) = A(u) = x$ for all nodes a in $p(u, x)$.
- Edge capacity constraints are met if reservations (edge e, time slot t) < $ec(e, t)$.
- Shelter capacity constraints are met if the number of nodes assigned to a shelter $s < sc(s)$.

Theorem 1

The ISA problem is NP-hard. Due to this article's size, the proof is provided elsewhere[1,6] by reducing a well-known NP-hard problem—that is, connected k-partitioning, to the ISA problem.

Figure 2 illustrates ISA's inputs and outputs with a transportation

```
Inputs:
    1) An undirected transportation graph G = (V, E), where V is a set of nodes and E is a set of edges
    2) Edge travel time t: E→Z+
    3) Edge capacity/unit time, ec: E→Z+
    4) Shelters S subset of V
    5) Shelter capacity sc: S→Z+

Output: Allotment A: V→S
Steps:
 1   while there are unallotted source nodes do
 2     Select the unallocated node with lowest travel time to nearest shelter accounting for edge capacity
       constraints and prior reservations but ignoring shelter capacity constraint.
 3     Increment load on nearest shelter (selected node).
 4     Update reservations on edges on shortestPath(u, nearestShelter(u)) for appropriate time slots to
       reflect new allotment
 5   end while
 6   Create a shelter graph SG = (S, SE). SG nodes represent individual shelters. Each SG node S_i is associated
     with weight w(S_i) = load(S_i)−sc(S_i), where load(S_i) is the number of nodes assigned to shelter S_i. A
     positive w(S_i) represents violation of the shelter capacity constraint. A negative w(S_i) represents shelter
     with available capacity. SG edges represent shelter pairs (S_i, S_j), such that there's an edge(u, v) in
     E with A(u) = S_i and A(v) = S_j.
 7   while a shelter capacity constraint is violated do
 8     Let S_x be an SG node with positive w(S_i), and S_y be an SG node with negative w(S_j).
 9     Find a directed path SP(S_x, S_y) in SG. If no path is founded, then return “no solution found.”
10     Let a directed path SP(S_x, S_y) in SG be S_x, . . . , S_i, S_i+1, . . . , S_y.
11     for each directed SG edge(S_i, S_i+1) in SP(S_x, S_y) do
12       Look up the set Q of directed edge(u, v) in E such that A(u) = S_i and A(v) = S_i+1.
13       Choose an edge(u, v) in Q such that reallocating shelter for node u to A(v) doesn't lead to spatial
         anomalies. (If no such edge in Q exists, then backtrack to Step 9 to consider alternative paths (S_x, S_y).)
14       Update loads on A(u) and A(v). Re-allocate shelter for node u to A(v).
15       Update reservations on edges on shortestPath(u, A(v)) for appropriate time slots to reflect new allotment.
16     end for
17   end while
```

Algorithm 1. The Crowd-separated Allotment of Routes, Exits, and Shelters (CARES) approach.

network of six source nodes, A–F, each with one evacuee and two shelter destinations (X, Y). Each shelter has capacity for three evacuees. For simplicity, edge travel time is one unit of time and capacity is one person per unit time. Figure 2b shows an allotment mimicking the NES paradigm (which is invalid because it exceeds the capacity of shelter Y).

Although NES avoids spatial anomalies, it violates shelter Y's capacity constraint, which can lead to congestion. Figure 2c shows an allotment that satisfies shelter capacity constraints but exhibits spatial anomalies, leading to potential movement conflicts due to noncompliance. In addition, service area contiguity is violated because C can't reach X without exiting the service area via red node B. This is also considered a spatial anomaly. Figure 2d shows a valid allotment that honors shelter capacity constraints, but it also eliminates spatial anomalies, which means it has no movement conflicts.

The Proposed Heuristic Algorithm

The CARES heuristic algorithm is based on the core idea of spatial anomaly avoidance.[1] Algorithm 1 shows an abstract description of the approach. In each iteration of the first **while** loop, the algorithm first identifies the unallotted node u with minimum travel time to the nearest shelter $ns(u)$, accounting for edge capacity constraints and prior reservations and ignoring shelter capacity constraint. It also makes an initial allotment of shelter $ns(u)$ to node u (Step 2), updating the load on $ns(u)$ (Step 3) and reserving edges along a shortest path from u to $ns(u)$ (Step 4). The loop terminates by creating an initial shelter allotment for each node in G.

However, the initial allocation might not meet some shelter capacity constraints. Step 6 creates a shelter-load-imbalance graph using shelters as nodes associated with weights to reflect any shelter capacity

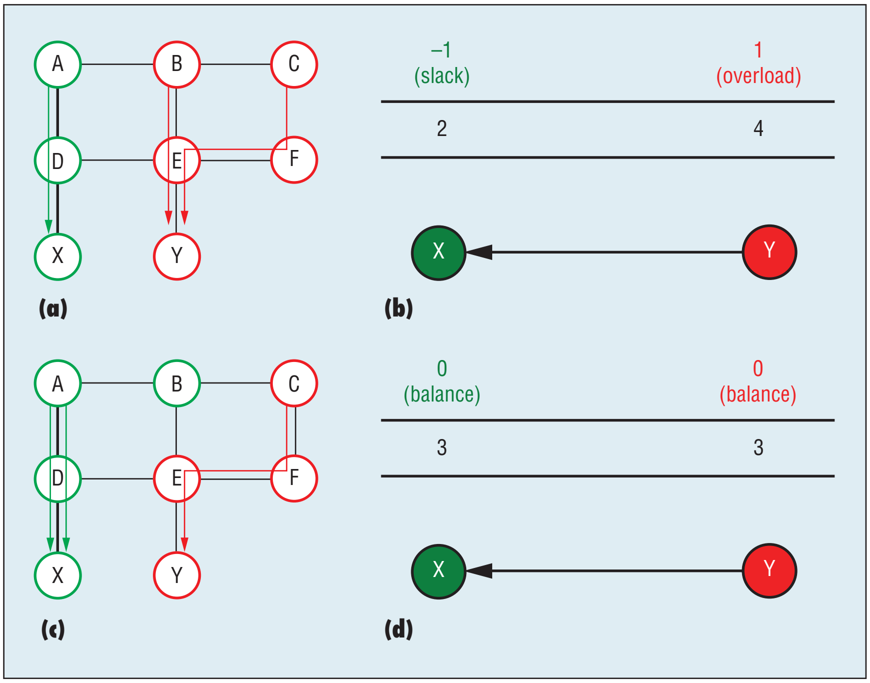


Figure 3. CARES: (a) a shelter allotment (first iteration), (b) a shelter graph (SG; first iteration), (c) a shelter allotment (second iteration), and (d) a updated shelter graph (SG; second iteration). Node Y is overloaded and violates a shelter capacity constraint, and shelter X has spare capacity.

constraint violations (or availability of spare capacity via negative weight values). Edges in the shelter-load-imbalance graph reflect opportunities for shelter re-allotments while avoiding spatial anomalies.

The second **while** loop tries to meet shelter capacity constraints—in each iteration, it first identifies an overloaded shelter S_x and a shelter S_y with spare capacity (Step 8) and then finds a path *SP* from S_x to S_y in the shelter-load-imbalance graph by identifying a sequence of shelters $S_x, \ldots, S_i, S_{i+1}, \ldots, S_y$ (Steps 9 and 10). If it can't find the *SP*, it returns "no solution found" (Step 9), at which point the user might choose to increase either shelter capacities (and the number of shelters) or transportation network size (or capacity) before invoking the algorithm. The **for each** loop (Step 11) moves one node from S_x to S_y via intermediate shelters in *SP* while avoiding two kinds of spatial anomalies. As noted earlier, the first represents a situation whereby a shelter assigned to a node is distinct from shelters assigned to that node's neighbors, whereas the second violates the shelter service area contiguity constraint. Avoidance of these spatial anomalies requires backtracking to Step 9 to explore alternative paths in the shelter-load-imbalance graph for re-allocating shelters.

Computational Complexity of CARES

To analyze the computational cost of CARES, let's assume G is the undirected transportation graph, u and v are nodes in G, (u,v) is an edge in G, n is the number of nodes in G, m is the number of edges in G, s is the number of shelters, p is the number of evacuees, *SG* is the shelter graph, S_x and S_y are shelters, and $SP(S_x,S_y)$ is a shortest path between S_x and S_y *in SG*. CARES starts by creating an initial allotment with the shortest path in G and updates reservations on path edges (Steps 1–5). Each iteration chooses the path for one group of people and reserves the capacities along the path (Step 4). In the worst case, each individual evacuee forms one group. Because the number of evacuees is p and the shortest path algorithm is $O(n \cdot \log n)$, these steps are $O(p \cdot n \cdot \log n)$. CARES then finds the excess and deficit nodes and computes the shortest path *SP* in *SG* (Steps 8 and 9) at a cost of $O(s \cdot \log s)$. Subsequently, it chooses an edge (u, v) (*SG* edge) and checks graph connectivity to find spatial anomalies (Step 13). If the node u is an articulation node, then service area of $A(u)$ isn't a connected graph after node u is removed from service area of $A(u)$. We can test this connectivity using a block-tree data structure.[4] We use a depth-first search (DFS) to create a block tree for every service area and test whether node u is an articulation node with the block tree. DFS takes $O(m)$ and contiguity checking for the service area after node removal takes $O(1)$.[4]

Because the number of service areas on a directed path, $SP(S_x, S_y)$, is bounded by $O(s)$, Step 13 takes $O(m + s)$. If one of the *SG* edges in the *SP* violates the connectivity constraint, CARES removes that edge from *SG*, backtracks to Step 9, and looks for another path *SP*. Because *SG* is a sparse graph, the number of *SG* edges is bounded by $O(s)$ and the number of backtracks is at most $O(s)$. After finding *SP*, CARES computes the corresponding path in G and updates reservations on the path edges, which takes $O(n \cdot \log n)$. The total number of iterations is bounded by $O(p)$ because the amount of excess load is at most $O(p)$. The time complexity of CARES is $O(p \cdot n \cdot \log n + p \cdot s \cdot (m + s \cdot \log s + n \cdot \log n))$. If we assume that $s << n$, then the complexity is $O(p \cdot s \cdot (n \cdot \log n))$.[4] Because the complexity is subquadratic in network size and number of evacuees, the algorithm may be considered scalable.

Figure 3 summarizes the execution trace of a CARES-generated shelter allotment. Steps 1–5 create an initial nearest-shelter-based allotment (Figure 3a). In Step 6, CARES creates the *SG* shown in Figure 3b and checks shelter capacity constraints. Figure 3b shows that *SG* node Y is overloaded and violates a shelter capacity constraint and that shelter X has spare capacity. To relieve the overload on Y, Steps 8–10 construct a path *SP*(X, Y) in *SG* from overloaded shelter Y to shelter X with spare capacity. In Step 12, CARES identifies edges (B, A) and (E, D), but Step 13 rules out edge (E, D) because re-allotting node E to shelter X leads to a spatial anomaly by violating shelter Y's service area contiguity. CARES selects edge (B, A) instead because re-allotting node B to shelter X doesn't lead to a spatial anomaly. Figure 3c shows the updated shelter allotment after the first iteration; Figure 3d shows that *SG* is balanced without any spatial anomaly, and the algorithm terminates.

Intelligent Strategies in CARES

CARES uses a time-aggregated graph[5] model to reserve specific combinations of edges and time slots for evacuees, avoiding edge time-sharing across service areas altogether. Therefore, if evacuees comply with the suggested routes, they don't criss-cross at any time.

CARES uses several intelligent strategies, including an iterative local search strategy (such as neighborhood search) to minimize the objective function over the feasible solution space, constraint satisfaction to preserve service area contiguity, and a network flow technique to achieve load balancing among shelters.[7,8]

To meet shelter capacity constraints, CARES first finds re-allotable nodes between two service areas and re-allots only a single node at a time according to an iterative improvement local search methodology. The key idea is to first select edges that connect two service areas and then find a single re-allotable incident (a node) of these edges (Step 13). This approach reduces the search space because it examines only the boundary edges between service areas. After the single node has been re-allotted, the two service areas smoothly expand (or shrink) towards meet service area capacity constraints.

Second, CARES applies the contiguity-constraint satisfaction technique and tests service area contiguity by inspecting a block tree (Step 13). Because the block tree detects an articulation node in constant time,[4] CARES tests the contiguity of service areas in $O(s)$ after node re-allotment on a directed path $SP(S_x, S_y)$. If the algorithm finds spatial anomalies during testing, it backtracks to Step 9 and looks for another $SP(S_x, S_y)$ that satisfies the contiguity constraint.

Finally, CARES updates the evacuation routes using a spatiotemporal network flow optimization technique and tries to balance the load among shelters. Step 7 checks the capacity for every shelter and Step 15 updates spatiotemporal network flows based on a time-aggregate graph. It's worth noting that the time-aggregate graph approach reduces the storage space with compact data structures.[5]

Experimental Evaluation

The main goal of experiments was to compare CARES algorithm with the state-of-the-art CCRP algorithm and the state-of-the-practice NES paradigm in meeting the aforementioned ISA challenges. A secondary goal was to evaluate the scalability of CARES, CCRP, and NES for different network size (n) and population (p).

Comparison Metrics

We used three metrics to measure the algorithms' ability to manage transportation choke points: evacuation time (time for the last evacuee to reach the allotted shelter), shelter arrival rate over the duration of evacuation, and cumulative percentage of the population reaching the shelters. For the case study described in the next section, we used visual maps to identify spatial anomalies—for example, evacuation groups whose allotted shelters differed from those of all of its neighbors—and assessed each algorithm's ability to avoid movement conflicts.

Simulation Description

We determined evacuation time, arrival time, and other comparison metrics for NES, CCRP, and CARES via iterative constant time interval simulations of the evacuation process. In each iteration, we determine the closest (source s, shelter destination d) pair along with the bottleneck capacity, say, c, of shortest path (s, d) in G. The movement of c evacuees from s to d along the selected path is simulated by making reservations on path edges.

Consider an edge e with one unit of travel time. If e is traversed with a starting time t_1, then *edge-available-capacity*(e, t_1) is reduced by c to simulate the movement. In addition, the number of evacuees at s is reduced by c and the number of evacuees at d is increased by c. The iterations end when the last evacuee reaches his or her assigned shelter.

Workload

Our experiments use a flash-flood scenario around the Jamarat complex in the tent city of Minna for Hajj in Makkah, Saudi Arabia. Overall, the Minna tent city is modeled with 16,043 nodes and 64,900 directed edges (average degree = 4.045) representing walkways, roads, ramps, and so on via OpenStreetMap. The average degree (undirected graph) is 2.02268, the normalized meshedness

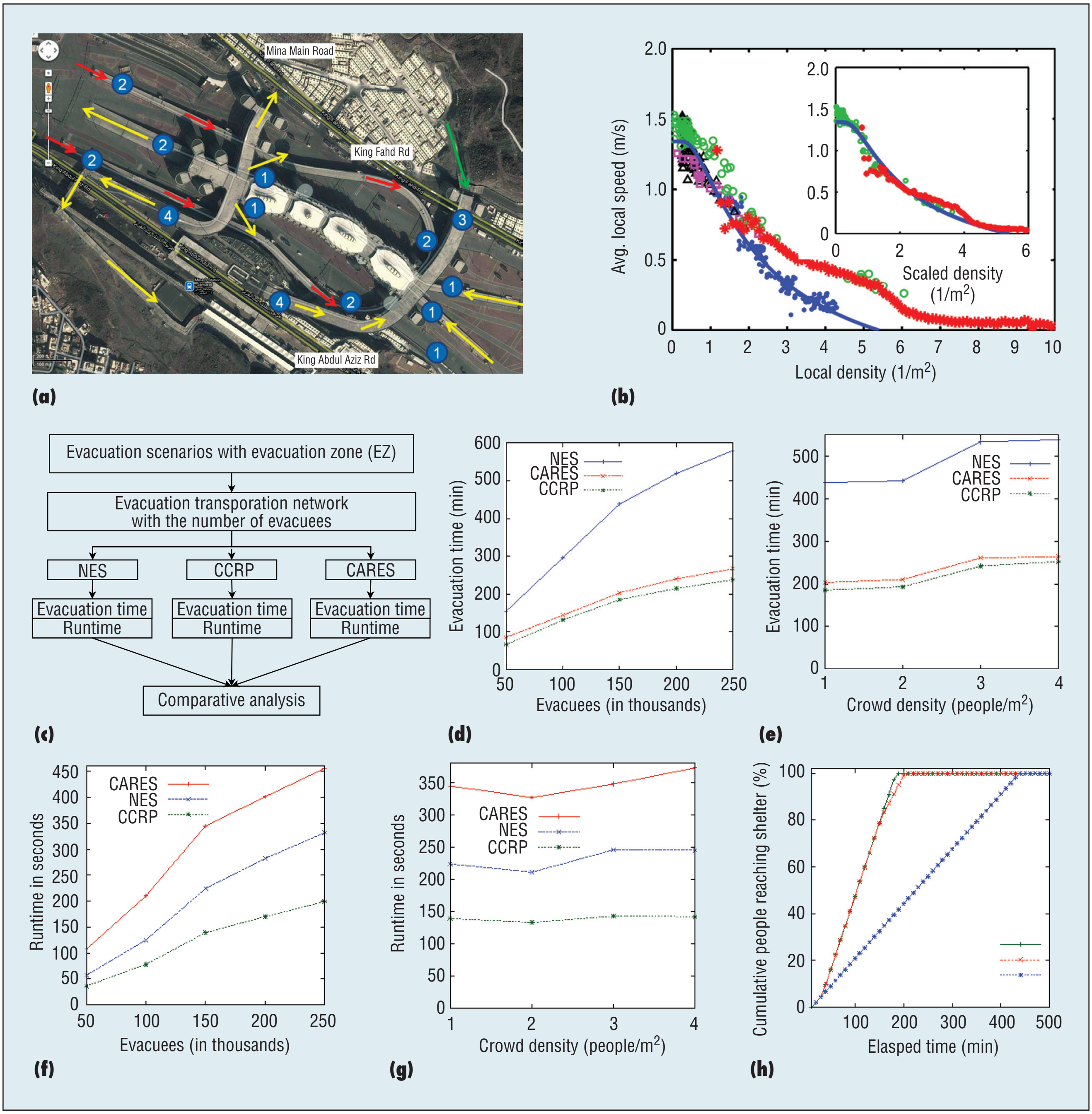


Figure 4. Experiment setup and results: (a) the Jamarat complex with multiple levels shown as 1, 2, 3, and 4 on associated ramps—every level is a distinct shelter destination; (b) crowd density and walking speed model of Hajj pilgrims[9]; (c) experiment design; (d) effect of the number of evacuees (1 person/m^2, n = 1,394); (e) effect of crowd density (150,000 evacuees, n = 1,394); (f) runtime comparison (1 person/m^2, n = 1,394); (g) runtime comparison (150,000 evacuees, n = 1,394); (h) cumulative evacuees of 150,000 evacuees (1 person/m^2, n = 1,394);

coefficient is 0.51145, and the average network efficiency is 0.07832.[10] The area affected by the flash flood is a subset of the tent city at the foothills near the Jamarat complex and includes 160 tent groups with 624 individual tents. These tent groups are linked by walkways that are approximately 2-m wide and connected by narrower walkways, ramps, roads (9-m-wide Souq Al Arab Road, Al jawharah Road, and King Faisal Road), and highways (11-m-wide King Fahd Road and King Abdul Aziz Road) to the Jamarat complex.

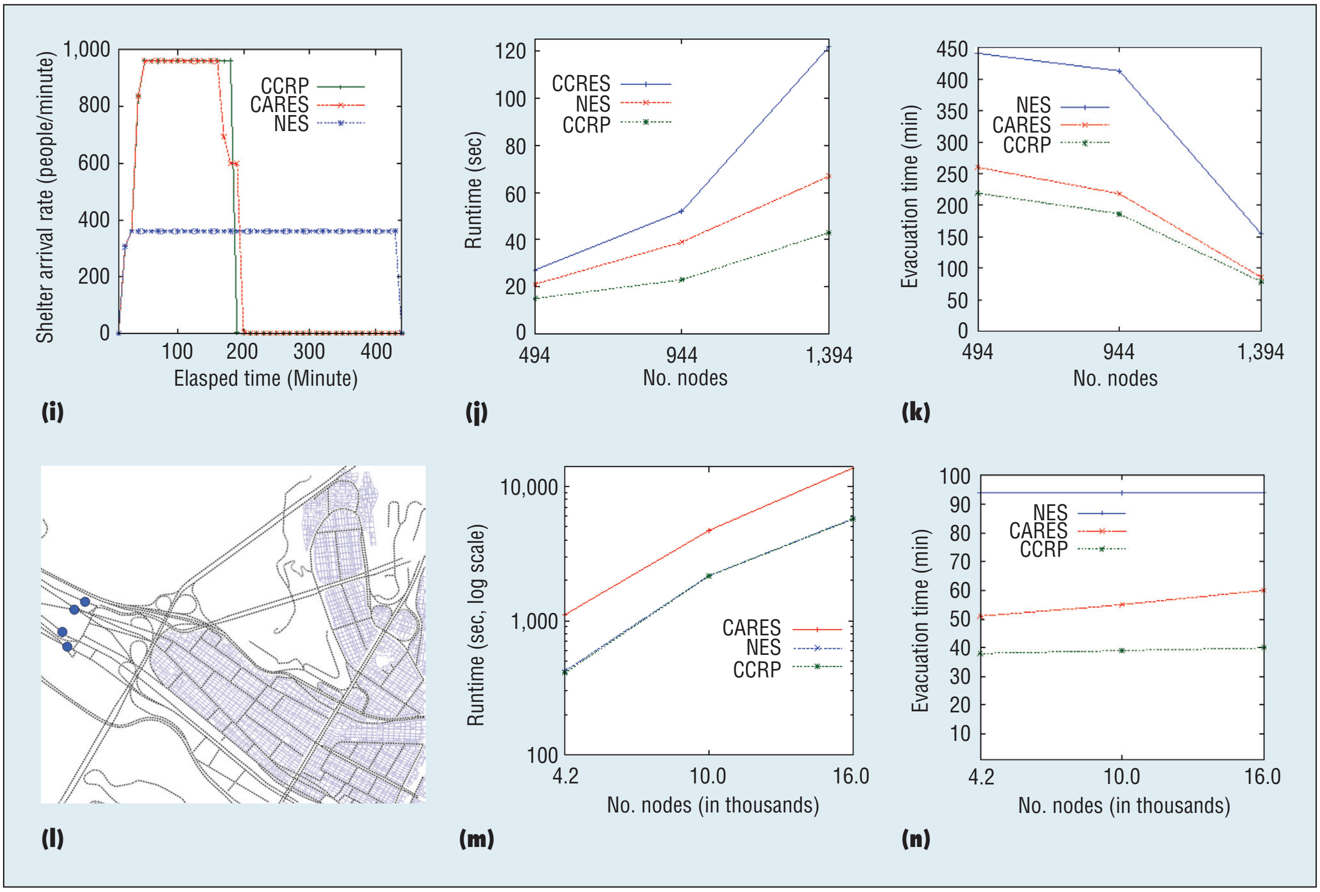


Figure 4. ***cont'd*** **(i) shelter arrival rate (person/minute; 150,000 evacuees, 1 person/m^2); (j) effect of number of nodes (50,000 evacuees, 1 person/m^2, *n* = 1,394); (k) effect of the number of evacuees (1 person/m^2, *n* = 1,394); (l) additional areas of Minna tents (16,043 nodes); (m) effect of number of nodes (runtime; 50,000 evacuees, 1 person/m^2, *n* = 16,043); and (n) effect of number of nodes (evacuation time; 50,000 evacuees, 1 person/m^2, *n* = 16,043).**

The subset of the tent city is composed of 1,394 nodes and 3,176 directed edges (average degree = 2.278). The destination shelters are defined by the two higher levels (levels 2 and 3) of the Jamarat complex structure, as shown in Figure 4a by red and green arrows.

Experiment Design

Variable parameters included the number of evacuees (50,000 to 250,000 in increments of 50,000), and the crowd density varying from 1 to 4 people/m^2. The walking speed was determined from crowd density by using previous studies of Makkah crowd movement (summarized in Figure 4b), along with ramp slope and elevator capacity. We implemented the NES, CCRP, and CARES algorithms in Java 1.7 with a 1-Gbyte memory runtime environment. All experiments were performed on an Intel Core i7-2670QM CPU running MS Windows 7 with 8 Gbytes of RAM. Figure 4c summarizes the overall experimental design.

Experimental Results

Figures 4d–4n depict experimental results that illustrate the effect of the number of evacuees and crowd density on computational cost (runtime), with the three metrics elucidating the ability of an algorithm to manage transportation choke points in terms of evacuation time, shelter arrival rates over the duration of the evacuation, and cumulative percentages of the population reaching the shelters.

Figures 4d and 4f show the effect of the number of evacuees on evacuation time and the computational cost of NES, CCRP, and CARES. Crowd density was fixed at 1 person/m^2, which corresponded to a walking speed of 1 m/sec.[9] For all algorithms, evacuation time increases with an increase in the evacuee population. Overall, CARES yields a considerably lower evacuation time but a higher computational cost relative to NES. However, CARES has a slightly longer evacuation time and a higher computational cost than CCRP due to its focus on avoiding movement conflicts.

Figures 4(e) and 4(g) show the effect of crowd density and associated walking speed on evacuation time and the computational cost of NES, CCRP and CARES. We chose 150,000 evacuees and varied crowd density from 1 to 4 people/m^2. The associated walking speed varied from 1 m/sec. down to 0.45 m/sec.[9] Entrance capacity of each shelter at level 2 was assumed to be 600 people/min., and 360 people/min. for level 3. For all algorithms, evacuation time increases as crowd density grows. Trends are similar to previous experiments, where CARES had much lower evacuation time but higher computational cost than NES. CARES has a slightly higher evacuation time and higher computational cost than CCRP due to avoidance of spatial anomalies.

Figures 4h and Figure 4i show the cumulative percentage of evacuees reaching shelters and the shelter arrival rates for NES, CCRP, and CARES. Overall, CCRP and CARES have similar performances and outperform NES, possibly because NES doesn't consider capacity constraints on routes and shelters. All algorithms show three phases of evacuation: pre-steady-state, steady-state, and post-steady-state. The steady-state corresponds to the flat shelter arrival rate in Figure 4i and the linear rate of shelter arrival in Figure 4h. The highest shelter arrival rate of both CCRP and CARES is 960 people/min. because they use both shelters (levels 2 and 3).

Figures 4j and 4k show the effect of the number of nodes in the network. We used the Minna area adjacent to the northern part of Jamarat, with 1,394 nodes and two shelters (levels 2 and 3; Figure 4a). We fixed the population size and reduced the network size—Figure 4j shows that as network size decreases, computational cost decreases, which is consistent with the complexity analysis given earlier. Figure 4k shows that the evacuation time increases as the number of nodes decreases. This is because congestion delay becomes high as the number of nodes decreases. Figures 4m and 4n show the effect of network size. In this case, we used the Minna area adjacent to the eastern part of Jamarat (Figure 4l), fixed the population size, and incrementally expanded the network size to the east, with 16,043 nodes and four shelters (shown as four blue circles). As the network size grows, the computational cost increases (Figure 4m), which remains polylog, as indicated by the analysis given earlier.

Managing 3 million Hajj pilgrims of various nationalities as well as cultural and lingual dispositions is a daunting challenge.

Discussion

A spatial anomaly doesn't slow down CCRP's evacuation and arrival times because it's a necessary but insufficient condition for the occurrence of spatiotemporal crisscrossing on various evacuation routes/road intersections. For example, a path might be used by different shelters in different time slots, leading to spatial anomalies, but this doesn't actually cause spatiotemporal crisscrossing (collision). However, spatial anomalies increase the risk of collision among groups of evacuees in transit if various flows of crowds arriving at intersections aren't properly monitored and paced. CCRP assumes 100 percent compliance for the recommended routes and a strict adherence to the transit schedule for all evacuees, including effective monitoring and pacing of evacuee flows at road intersections. Using these assumptions, CCRP reduces the risk of collisions and apparently provides a faster solution. However, in the real world, compliance assumptions might not be achievable without a large-scale deployment of first responders. Lack of such resources could lead to substantial degradation of CCRP performance in terms of slowing down the evacuation process and increasing evacuation and arrival times. CARES, on the other hand, generates a solution that's free from spatial anomalies to begin with and hence doesn't require substantial resources for monitoring and pacing flows.

Case Study in a Hajj Scenario

Managing 3 million Hajj pilgrims of various nationalities as well as cultural and lingual dispositions is a daunting challenge. Numerous incidents and causalities during Hajj have been covered by media.

Flash-Flood Scenario

Minna is a rough mountain area southeast of Makkah; it's a narrow valley about 6 km from the Sacred Mosque of Makkah on the way to Arafat. It covers approximately 812 hectares, 52 percent of which is flat land. Pilgrims stay in Minna for at least four days and three nights as part of the Hajj rituals.

A possible shelter to evacuate pilgrims during a flash flood is the Jamarat complex, five-level structure served by ramps connecting each level with one-way flow. Movement toward Azizyah/Haram is allowed, but cross movements are restricted. Various levels are connected for emergency movements. Out of four days in Minna, Jamarat can be used for relief

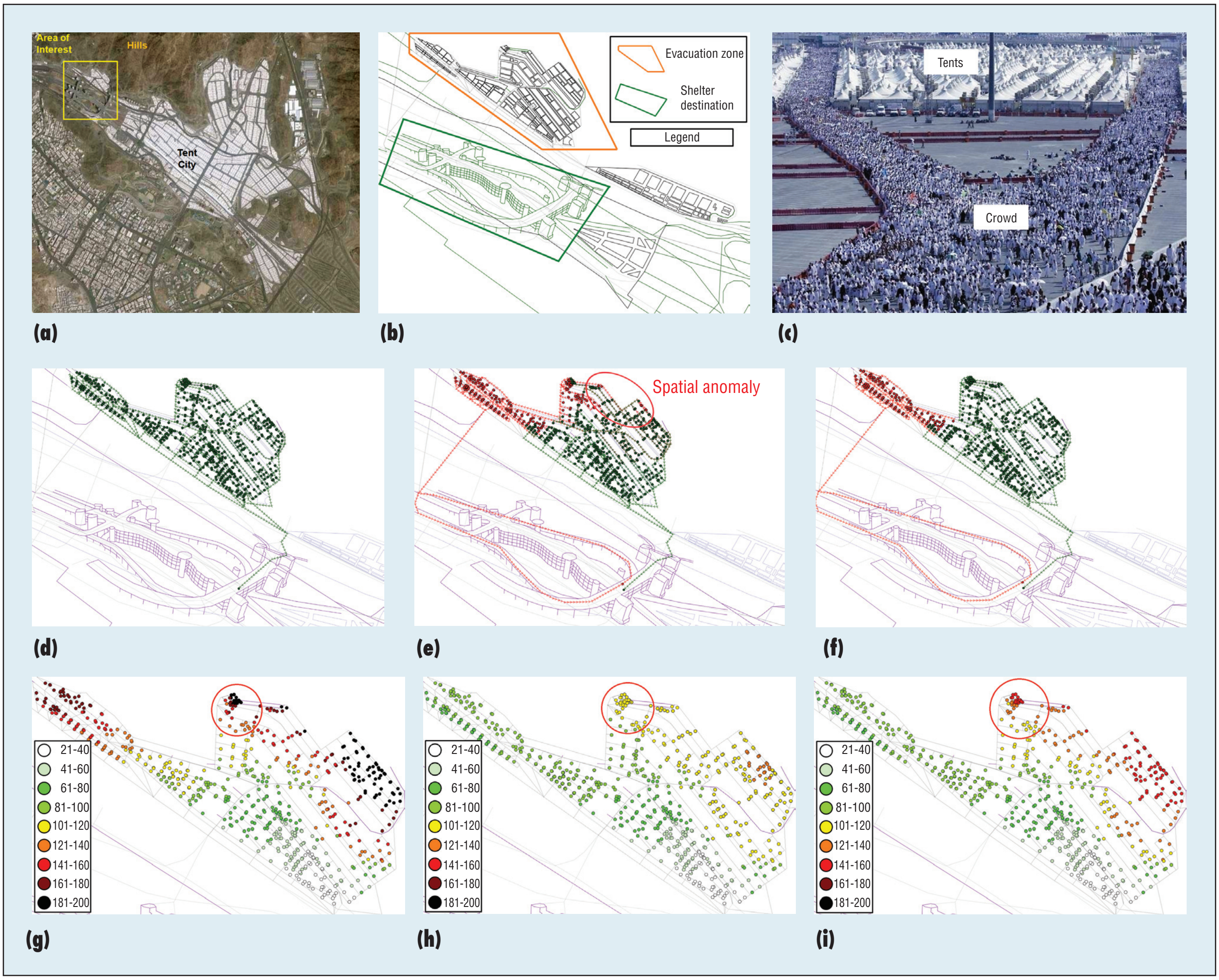


Figure 5. Case study in a Hajj scenario. The shelter allotment and shelter arrival time distribution for three algorithms, for 50,000 evacuees, 4 people/m^2, evacuation unit = tent: (a) area of interest within tent city, (b) evacuation zone and shelter destination (Jamarat complex) in area of interests, (c) crowd walking from tent city to Jamarat complex (www.flickr.com), (d) NES allotment, (e) CCRP allotment, (f) CARES allotment, (g) NES arrival time, (h) CCRP arrival time, and (i) CARES arrival time.

activities one day only (day 8 of Dhul-Hijjah), as a staging place for victims. For the remaining three days (days 10–12 of Dhul-Hijjah), it's heavily crowded and can only serve as a transit point for evacuation by metro, bus, or helicopter.

Figure 5a shows 6,000 Minna tents and 20 square km area of interest for the flash-flood scenario. Figure 5b shows the evacuation zone for the tent city (shown in orange) and the shelter destination of the Jamarat complex (shown in green). In our analysis, we chose 160 tent groups over an approximately 1-km area. We assigned 50,000 evacuees to this area and used 4 people/m^2 crowd density.

Movement Conflict Comparison

Figures 5d–5f show the shelter allotments provided by NES, CCRP, and CARES. Red shelter is on level 2 and green shelter is on level 3 of the Jamarat complex, with tents and routes colored by shelter destinations. The CCRP allotment exhibits spatial anomalies, as highlighted in Figure 5e via red circles indicating movement conflict risks due to non-compliance. In contrast, NES and CARES shelter allotments show no spatial anomalies. Figure 5d shows that NES uses only one shelter (level 3) due to its proximity to all source nodes. This could explain the longer evacuation time for NES (Figure 4).

Visualizing Spatial Distribution of Shelter Arrival Time

Figures 5g–5i show the spatial distribution of shelter arrival time

THE AUTHORS

KwangSoo Yang is an assistant professor in the Department of Computer and Electrical Engineering and Computer Science at Florida Atlantic University. His research interests include spatiotemporal network databases, spatiotemporal networks, and evacuation routing problems. Yang has a PhD in computer science from the University of Minnesota. Contact him at ksyang@cs.umn.edu.

Apurv Hirsh Shekhar is an undergraduate student at Johns Hopkins University. His research interests include spatiotemporal networks, computational bioengineering, and computational medicine. He received the 1st Place Individual Award in Computer Science from the IEEE Computer Society at the 2013 Intel International Science and Engineering Fair. Contact him at ahshekhar@jhu.edu.

Faizan Ur Rehman is a PhD candidate in the Department of Computer Science at University of Grenoble Alpes, France; he also works as a research assistant in Umm Al-Qura University's Center of Research Excellence in Transportation and Crowd Management. His research interests include spatiotemporal databases, algorithms, and multimedia. Contact him at fsrehman@uqu.edu.sa.

Hatim Lahza is a research assistant on an emergency management research project at Umm Al-Qura University. His research interests include spatial database, data mining, and software engineering. Lahza has a BS in computer science from Umm Al-Qura University. Contact him at hfmlahza@uqu.edu.sa.

Saleh Basalamah is an associate professor and deputy director of the GIS Technology Innovation Center at Umm Al-Qura University. His research interests include spatial systems and multimedia. Basalamah has a PhD in bioengineering from Imperial College London. He's a senior member of IEEE and a member of ACM. Contact him at smbasalamah@uqu.edu.sa.

Shashi Shekhar is a McKnight Distinguished University Professor at University of Minnesota. His research interests include spatial databases, spatial data mining, and geographic information systems. Shekhar has a PhD in computer science from the University of California, Berkeley. He's a fellow of IEEE and the AAAS. Contact him at shekhar@cs.umn.edu.

Imtiaz Ahmed is a professor at Umm Al-Qura University. His research interests include sustainable transportation and disaster risk management. Ahmed has a PhD in civil engineering from Purdue University. He's a member of the Pakistan Engineering Council and the Institution of Engineers Pakistan. Contact him at iaahmed@uqu.edu.sa.

Arif Ghafoor is a professor in the School of Electrical and Computer Engineering at Purdue University. His research interests include information security and distributed multimedia systems. Ghafoor has a PhD in electrical engineering from Columbia University. He's a fellow of IEEE. Contact him at ghafoor@purdue.edu.

using color coding (darker colors indicate later arrival times) for NES, CCRP, and CARES. All three allotments show that the area highlighted with red circles at the top center has the highest values for shelter arrival time, despite its proximity to foothills and flash floods. If a shelter's arrival time is unacceptable, authorities could use additional routes to speed up evacuations. Note that, for many tents, NES yields a high evacuation time relative to CCRP and CARES.

As mentioned earlier, CARES is meant for preplanning under various evacuation scenarios and exhibits little adaptability during evacuation. However, its sub-quadratic time complexity along with a real-time crowd monitoring methodology for extrapolating crowd density can be exploited for real-time applications. Such research can be pursued in future.

Acknowledgments

This research was supported by the Center of Research Excellence in Transportation and Crowd Management, Umm Al-Qura University, Saudi Arabia, and the US National Science Foundation.

References

1. K. Yang et al., "Intelligent Shelter Allotment for Emergency Evacuation Planning: A Case Study of Makkah," HajjCore tech. report P1104-T1, 2012; http://docs.lib.purdue.edu/cctech/9.
2. M. Ben-Akiva et al., "Development of a Deployable Real-Time Dynamic Traffic Assignment System: Dynamit and Dynamit-P Users Guide," Intelligent Transportation Systems Program, MIT, 2002
3. M. Skutella, "An Introduction to Network Flows over Time," *Research Trends in Combinatorial Optimization*, Springer, 2009, pp. 451–482.
4. B. Korte et al., *Combinatorial Optimization*, Springer, 2012.
5. S. Shekhar et al., "Experiences with Evacuation Route Planning Algorithms," *Int'l J. Geographical Information Science*, vol. 26, no. 12, 2012, pp. 2253–2265.
6. K. Yang et al., "Capacity-Constrained Network-Voronoi Diagram: A Summary of Results," *Proc. 13th Int'l Conf. Advances in Spatial and Temporal Databases*, 2013, pp. 56–73.
7. V. Kumar, "Algorithms for Constraint-Satisfaction Problems: A Survey," *AI Magazine*, vol. 13, no. 1, 1992, p. 32.
8. A. Nareyek et al., "Constraints and AI Planning," *IEEE Intelligent Systems*, vol. 20, no. 2, 2005, pp. 62–72.
9. D. Helbing et al., "Dynamics of Crowd Disasters: An Empirical Study," *Physical Rev. E*, vol. 75, no. 4, 2007, p. 046109.
10. J. Buhl et al., "Topological Patterns in Street Networks of Self-Organized Urban Settlements," *European Physical J. B: Condensed Matter and Complex Systems*, vol. 49, no. 4, 2006, pp. 513–522.