A Framework for Emergency-Evacuation Planning Using GIS and DSS

Reham Ebada Mohamed, Essam Kosba, Khaled Mahar and Saleh Mesbah

Abstract There has been a growing need for the use of information and decision-making systems in evacuation planning as a part of emergency management in order to reduce as, many losses as possible. To minimize damage, an accurate and effective evacuation plan that gives more than one evacuation path considering the changing road conditions in minimal time is imperative. The use of Geographic Information Systems (GIS), Decision-Support Systems (DSS), and shortest-path algorithms as a solution for this problem is the subject of this chapter. A framework for providing preparedness and response plans after an emergency event occurs is proposed. The plan provided by the proposed framework includes a calculated degree of hazard posed by that event (three emergency models are incorporated), the radius of affected area (a spatial overlay to draw a buffer zone is conducted), identification of all safe destinations, and the best alternative paths for evacuation from inside the buffer zone to safe destinations based on the dynamic road conditions displayed on the map. To identify all of the safe destinations and obtain the best alternatives, a graph theory–based model is proposed based on a developed algorithm to get everyone the closest safe destinations. Dijkstra's algorithm is executed from a single source inside the buffer to all identified safe destinations resulting in the minimum travel time path and other alternative paths displayed on the map. To evaluate the proposed framework, a GIS-based evacuation-planning (G-BEP) prototype is implemented. The prototype is tested with different emergency types, different variables, and different street maps. The prototype is also found to respond differently based on the dynamic road conditions.

R.E. Mohamed (✉) · E. Kosba · K. Mahar · S. Mesbah
College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport (AASTMT), Alexandria, Egypt
e-mail: reham.ebada@gmail.com

E. Kosba
e-mail: ekosba@aast.edu

K. Mahar
e-mail: khmahar@aast.edu

S. Mesbah
e-mail: saleh.mesbah@gmail.com

V. Popovich et al. (eds.), *Information Fusion and Intelligent Geographic Information Systems (IF&IGIS'17)*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-59539-9_16

Keywords GIS-based DSS · Evacuation planning · Emergency models · Dijkstra's shortest path · Evacuation alternative paths · Dynamic road conditions · Safe destinations

1 Introduction

An emergency event is uncertain, sudden, and complex for analysis. Emergencies are either caused by a natural disaster or a deliberate act. Natural disasters include earthquakes, floods, fires, tsunamis, tornadoes, hurricanes, volcanic eruptions, or landslides. Deliberate acts include explosions, chemical spills, radiation, terrorism attacks, and others.

In recent years, the world has experienced many crises related to disaster and emergency occurrences. Those crises usually cause a great loss of population and wealth. It has become imperative to establish a scientific and effective plan to manage the disaster and to reduce losses as much as possible. Such plans are the fundamental purpose of emergency management in order to evacuate as many victims as possible in the shortest time. The essential mapping of the spatial relationships between natural-hazard phenomena (earthquakes, fires, cyclones, etc.) and the elements at risk (people, buildings, and infrastructure) require the use of tools such as GIS. The most significant relationships in risk analysis and modeling are largely spatial, and although GIS did play an earlier role, but the plans were relatively primitive, rarely commercially available, and often experimental or research-focused rather than operational.

The use of DSS would improve the efficiency and effectiveness of evacuation. The focus on emergency management, especially the evacuation-planning field, is to prevent disasters—or mitigate them when they occur—as quickly as possible. Accurate and reliable information and spatial data on disaster, as well as how to quickly deal with the statistical summary and analysis, requires efficiency and effectiveness; thus, the use of DSS is required.

This chapter is an extended work of a previously published paper by the same authors [1]. This work focuses on using GIS for obtaining information about the emergency event, calculating its danger rating, performing buffer analysis and overlay, identifying the closest safe destinations, and solving the shortest path (based on Dijkstra's model) to obtain the best evacuation path and other alternatives based on travel time. Despite the abundance of research in this area, most research papers calculate the best optimal path to our human knowledge. Compared with related work presented in this chapter, one of its contributions is presented as the second-best and third-best alternative paths rather than calculating one optimal evacuation path.

The reminder of this chapter is organized as follows: related work is presented in Sect. 2, which covers related studies of evacuation planning using GIS, DSS, and graph-theory algorithms. Section 3 covers the methodology showing the proposed framework, the parameters used to select the best paths, the developed algorithms, and the emergency models and equations. Section 4 presents the implementation of a G-BEP prototype and the results by covering the functionality of buffer analysis and overlay as well as evacuation-path generation. Finally, the conclusion and future work are presented in Sect. 5.

2 Literature Review and Related Work

The value of GIS in emergency management arises directly from the benefits of integrating a technology designed to support spatial decision-making into a field with a strong need to address numerous critical spatial decisions [2]. An important step in examining the role of GIS in emergency management is selecting a conceptual framework to help organize the existing research-and-development activities. One such framework is comprehensive emergency management (CEM) [3]. The use of GIS in providing the platform for CEM is discussed in [4].

CEM activities can be grouped into four phases; however, in examining the GIS literature, it is more appropriate to reduce them into three phases: (1) mitigation, (2) preparedness and response, and (3) recovery. This is because many GIS developed in the preparedness phase are used in the response phase [2]. Challenges for GIS in emergency preparedness and response are presented in [5].

Studies related to emergency response based on GIS, computational-simulation models, mathematical-assessment models, and dynamic algorithm to obtain the optimal solution are presented in [6–9]. Other studies focus not only on natural phenomena but also on traffic accidents [10, 11].

Some studies have focused based on GIS and DSS for evacuating large-scale buildings or public squares [12, 13]. Other studies focus on real-time evacuation systems in open areas based on GIS and DSS such as the Blue-Arrow system [14] and the emergency decision-making system developed in [15]. Shortcomings of the Blue-Arrow system are that target shelters should be pre-defined in the system because the radius of affected area might not be known to the user, and the system does not consider road conditions. The shortcoming in [15] is that it does not provide other alternatives to the evacuation path.

An improved Dijkstra's algorithm applicable to the GIS drilling accident–emergency rescue system is proposed in [16]. The shortcoming of the system is that the search for the evacuation path is based on defining a source and a single destination node. A dynamic vehicle routing during a peak-hour traffic system is implemented in [17] using Krushkal's algorithm to obtain the best route based on different parameters, but its shortcoming is that it only generates the optimum path.

Such studies contributed as a motivation to contemplate this research to overcome the mentioned shortcomings by integrating some graph-theory algorithms with GIS and DSS to compute the best alternative evacuation paths.

3 Methodology

This section presents an explanation of the proposed framework including a description of each component and the chosen algorithms. Additionally, it discusses how the framework adopts both the previously mentioned issues that must be considered regarding the development and limitations of previous work. Moreover, it covers the background of the used emergency models.

3.1 G-BEP: The Proposed Framework

G-BEP is proposed and developed in this chapter. It overcomes the shortcomings of previous research by the calculating radius of affected area and providing preparedness and response plans after the incident has occurred. Along with its degree of hazard, it identifies all safe destinations (algorithm), obtains alternative evacuation paths (not just the best one), and considers dynamic road-condition data that contribute as parameters for the selection of the best alternative paths. Figure 1 shows the main framework components of G-BEP. Note that developed components appear in the shaded area.

G-BEP is based on using GIS for obtaining information about the emergency event, calculating its danger rating, and performing buffer analysis and overlay on a street map in a Geo-Database. Three emergency types are represented: a fire event, a hurricane event, or an earthquake event. The spread area is calculated for each event based on three emergency models: the McArthur Fire Danger Index Model, the Carvill Hurricane Index (CHI), and the Earthquake Damage Estimation, respectively.

To obtain the best alternative, a graph theory-based model to obtain the shortest path (based on Dijkstra's) is developed in G-BEP, and alternatives based on the travel time are generated. For this to be applicable, the original street map with the

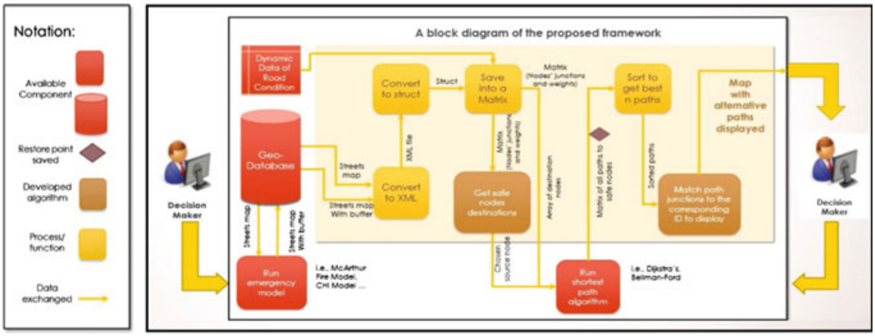

Notation:
Available Component
Restore point saved
Developed algorithm
Process/ function
Data exchanged
A block diagram of the proposed framework
Dynamic Data of Road Condition
Convert to struct
Struct
Save into a Matrix
Matrix (Nodes' junctions and weights)
Sort to get best n paths
Map with alternative paths displayed
Geo-Database
Streets map
Streets map With buffer
XML file
Convert to XML
Matrix (Nodes' junctions and weights)
Get safe nodes destinations
Array of destination nodes
Matrix of all paths to safe nodes
Sorted paths
Match path junctions to the corresponding ID to display
Decision Maker
Streets map
Streets map With buffer
Run emergency model
I.e., McArthur Fire Model, CHI Model ...
Chosen source node
Run shortest path algorithm
I.e., Dijkstra's, Bellman-Ford
Decision Maker

Fig. 1 G-BEP framework

drawn buffer zone must be converted to a graph showing streets as edges, street junctions as nodes, and the travel time between each node as an edge weight or cost. This is performed by executing some processes and functions that appear in the shaded area in the framework. The two developed algorithms appearing in that area are explained in the third sub-section of this section. Dijkstra's algorithm is chosen to be the shortest-path algorithm used in this research because according to [18, 19], it has been recognized as the best algorithm for the shortest-path problem since 1959.

After obtaining a result, the decision-maker commands a rescue team on the emergency spot with an evacuee vehicle through the generated paths; in that case, some roads' density would reach its maximum value: The best paths generated at a moment are not necessarily the best paths in the moment after. In addition to the traffic flow, other dynamic parameters could exist (these parameters are elaborated in the next section). Thus, dynamic data of the road conditions should be obtained and read into the framework so that weights/costs of the edges in the graph are recalculated and the result is updated. Feedback from the decision-maker would be a different time input for the rate of spread or a new decision-making process based on the dynamic data of the road conditions.

3.2 Selecting the Best Path

Some particular parameters contribute to selecting the best path: the current traffic flow, the safety of the road, and the existence of sudden blocks. The first parameter, traffic flow, determines the mean speed at any moment. The flow is calculated by knowing the current density of the road traffic. The number of lanes of the road and the road's capacity determine the number of vehicles passing a road at the certain time at which the flow is being measured. To calculate the flow, equations in [20] state the relationship between numbers of vehicles, length of the road, road density, and traffic flow. When the density reaches a maximum jam density, flow must be zero because vehicles will line up end to end. Thus, the road with a flow equals to zero must be excluded because its capacity will not allow extra new vehicles.

After obtaining the dynamic data of the road conditions, the new speed based on the traffic flow is updated and considered as new weights for the edges in the built graph. If the flow equals zero, the speed equals zero as well; thus, the travel time will equal infinity. In that case, the edge of that weight/cost is a nonexistent edge for the shortest-path algorithm because it searches for edges with minimal weights. Selection is based on obtaining the fastest route with the minimum travel time and other alternatives.

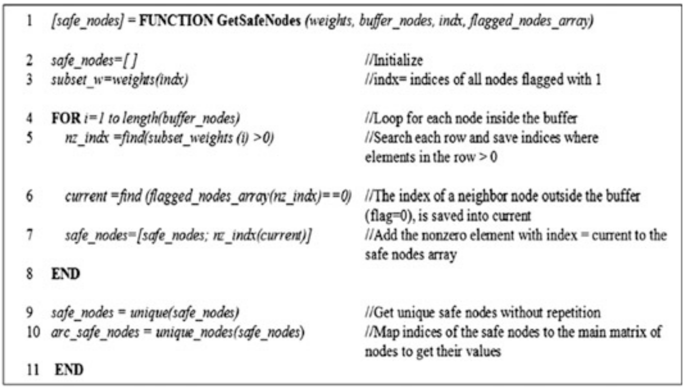
The second parameter, safety of the road, means that road's conditions must be considered safe, i.e., has the emergency's hazard already affected the road? If any road is affected by the emergency, it should not be considered as an existent edge for the decision maker to select as a potential alternative. Thus, as mentioned in the previous point, the weight/cost of that edge must equal to infinity.

The third parameter is the existence of any sudden blocks in the road. This means that if vehicles are already guided to select a road, a sudden block due to the emergency's consequences might appear at any segment of that road. In that case, all vehicles lined in that road should be moved back from that road to the nearest exit to other road. In that case, no new vehicles shall enter that road while the other vehicles are moving back, nor shall the road be considered as a potential alternative in the following decision-making process. Thus, the weight/cost of that edge must equal to infinity as well.

3.3 Developed Algorithms in G-BEP

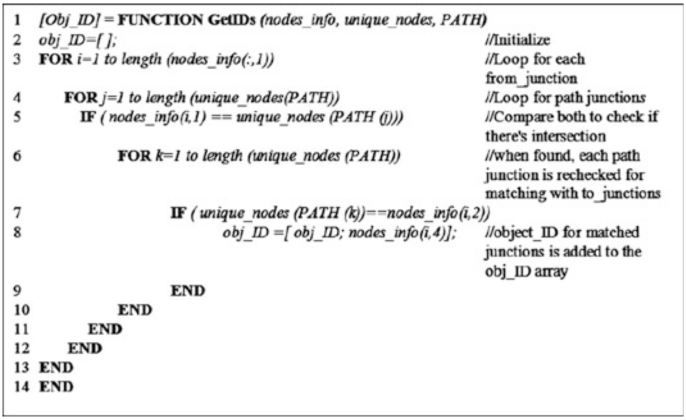
The major feature of G-BEP is developing two algorithms (the brown squares in Fig. 1). One of them is for identifying safe destinations (mentioned in step 7), which are the closest safe nodes outside the buffer's boundary, which are sent to the shortest-path algorithm as destinations; the pseudo-code is shown in Fig. 2.

The algorithm is run after an overlay analysis (an intersect operation) is performed for the streets map after generating a buffer zone of the affected area. Some nodes and edges are inside the buffer, and some nodes and edges are outside its boundary. All of the features inside the buffer zones boundary that intersect with the outer graph are saved as a new feature. The algorithm marks nodes' junctions connecting those edges with a flag = 1. All other nodes outside the boundary take a zero flag. The algorithm visits each node inside the buffer, checks the flag of all edges of its neighbors, and then it checks neighbor nodes. If the algorithm finds no flag with value equaling 0, no safe node has been found yet. Looping on all nodes inside the buffer, the algorithm visits one of the neighbors of a last visited node. After checking the flags, if the algorithm finds one neighbor with a different flag, which is a safe node, it is saved into the safe_nodes array. The same process applies



```
[safe_nodes] = FUNCTION GetSafeNodes (weights, buffer_nodes, indx, flagged_nodes_array)

safe_nodes=[ ]                                       //Initialize
subset_w=weights(indx)                               //indx= indices of all nodes flagged with 1

FOR i=1 to length(buffer_nodes)                      //Loop for each node inside the buffer
   nz_indx =find(subset_weights (i) >0)              //Search each row and save indices where
                                                     elements in the row > 0

   current =find (flagged_nodes_array(nz_indx)==0)   //The index of a neighbor node outside the buffer
                                                     (flag=0), is saved into current
   safe_nodes=[safe_nodes; nz_indx(current)]         //Add the nonzero element with index = current to the
                                                     safe nodes array
END

safe_nodes = unique(safe_nodes)                      //Get unique safe nodes without repetition
arc_safe_nodes = unique_nodes(safe_nodes)            //Map indices of the safe nodes to the main matrix of
                                                     nodes to get their values
 END
```

Fig. 2 Identifying the safe-destinations pseudo-code



```
[Obj_ID] = FUNCTION GetIDs (nodes_info, unique_nodes, PATH)
obj_ID=[ ];                                                    //Initialize
FOR i=1 to length (nodes_info(:,1))                            //Loop for each
                                                               from_junction
    FOR j=1 to length (unique_nodes(PATH))                     //Loop for path junctions
      IF ( nodes_info(i,1) == unique_nodes (PATH (j)))         //Compare both to check if
                                                               there's intersection
          FOR k=1 to length (unique_nodes (PATH))              //when found, each path
                                                               junction is rechecked for
                                                               matching with to_junctions
              IF ( unique_nodes (PATH (k))==nodes_info(i,2))
                  obj_ID =[ obj_ID; nodes_info(i,4)];          //object_ID for matched
                                                               junctions is added to the
                                                               obj_ID array
              END
          END
        END
    END
END
END
```

Fig. 3 Matching the path junctions with the object_ID pseudo-code

to all other nodes inside the buffer, and all safe nodes are saved. The safe nodes, which are junctions and edges connected to these nodes, represent the end of every path that will be found by the shortest-path algorithm from a source node to any of these safe-node destinations.

The second algorithm is for matching the path junctions (a path is the output of the shortest-path algorithm) with the saved streets' junctions to obtain the corresponding object_ID, which is sent to the map show the function to display the generated path. The pseudo-code is shown in Fig. 3.

The algorithm matches each value appearing in path with the saved junctions to obtain the ID when an intersection between a path and an (F_Junc, T_Junc) is found. It begins with searching F_Junc to match each element with all the elements in path.

Once it finds a matching element in F_Junc, it checks the value of T_Junc for the same element. Next, it searches if the same value in T_Junc appears in path as well. It loops from the beginning of PATH until a matching element is found. When an F_Junc paired with a T_Junc appears in the path, the algorithm saves the ID of the corresponding junctions.

3.4 Emergency Models and Equations

The models presented are used in the framework to determine the degree of hazard posed by the emergency event and to calculate the buffer of the affected area.

McArthur Fire Danger Index Model

The criteria of choosing a fire-danger rating are the area of validity, the availability of data for the model input, and the basis of the model. The chosen model is the

McArthur Forest Fire Danger Index. It works on open conditions on flat ground; input data are available; and it is an empirical model. Certain equations are added to the G-BEP system to calculate the Fire Danger Index (FDI), thus indicating which Fire Danger Rating (FDR) the fire has generated, and FROS determines the radius of affected area (buffer zone). The equations used to calculate the rate of spread on which the buffer zone area is defined are as follows [21, 22]:

$$\mathrm{FDI} = 2.0(10A)^{0.987}\mathrm{e}^{(-0.450+0.0338T-0.0345H+0.0234V)} \tag{1}$$

$$\mathrm{FROS} = 0.0012\mathrm{FDI} \times \mathrm{FL} \times \mathrm{e}^{(0.069S)} \tag{2}$$

where FDI is the fire danger index; A is the fuel-availability index (numerical value ranges from 0 to 1); T is the air temperature (°C); H is the relative humidity (%); V is the average wind velocity at a height of 10 m (km/h); FROS is the forward rate of spread (km/hr); FL is the fuel weight (t/ha); and S is the ground slope (degrees). Valid ranges include $A \in (0, 1)$, $T \in (5, 45)$, $H \in (5, 70)$, and $V \in (0, 65)$.

Carvill Hurricane Index

An equation to calculate CHI is added to G-BEP to calculate the buffer zone and output the hurricane index to the user, thus indicating the danger ranking of the hurricane according to inputs added by the G-BEP user. The CHI is calculated as follows [23]:

$$\mathrm{CHI} = \left(\frac{V}{V_0}\right)^3 + \frac{2}{3}\frac{R}{R_0}\left(\frac{V}{V_0}\right)^2 \tag{3}$$

where V is the maximum sustained wind speed of a hurricane in miles per hour; V_0 is constant and equals 74 mph; R is the radius of hurricane force winds of a hurricane in miles (i.e., how far from the center of hurricane winds ≥ 74 mph or greater are experienced); and R_0 is the radius of hurricane force winds at 74 mph, which equals 60 miles.

Earthquake Damage Estimation

In order to add an earthquake modeling feature to G-BEP system, it is required to have data regarding how far the damage of an earthquake may reach based on its magnitude. The radius of affected area as found by [24] estimated based on the Richter magnitude as follows (the buffer zone is based on the earthquake-magnitude input):

- Magnitude 5.0–6.0 = 2.36–6.14 km
- Magnitude 6.0–7.0 = 6.14–15.92 km
- Magnitude 7.0–8.0 = 15.92–41.30 km
- Magnitudes >8.6 = >41.30 km.

4 Implementation and Results

To evaluate the proposed framework explained in Chapter "Fuzzy-Vector Structures for Transient-Phenomenon Representation", the G-BEP prototype is implemented, and the results are displayed and compared. The emergency model is incorporated into the proposed solution by developing a new toolbar called Evacuation-Analysis Tool, which is easily added by the user into the program inside ArcMap®. The shortest-route analysis is carried out after performing the buffer analysis and displaying the buffer zone on the map, thus showing to the user the calculated affected zone of any previously added one of the three emergency events. The shortest-route analysis is performed in the second feature.

4.1 G-BEP Functionality of Buffer Analysis and Overlay

Using the newly developed Evacuation-Analysis Toolbar, shown in Fig. 4, the user can choose to add the event's data.

According to the mathematical model used, the model calculates two outputs: (1) the degree of hazard posed by the emergency event and (2) the radius of affected area, which in turn is sent to the ArcObjects® function that creates a buffer layer with the same projection and displays the buffer on the streets map.

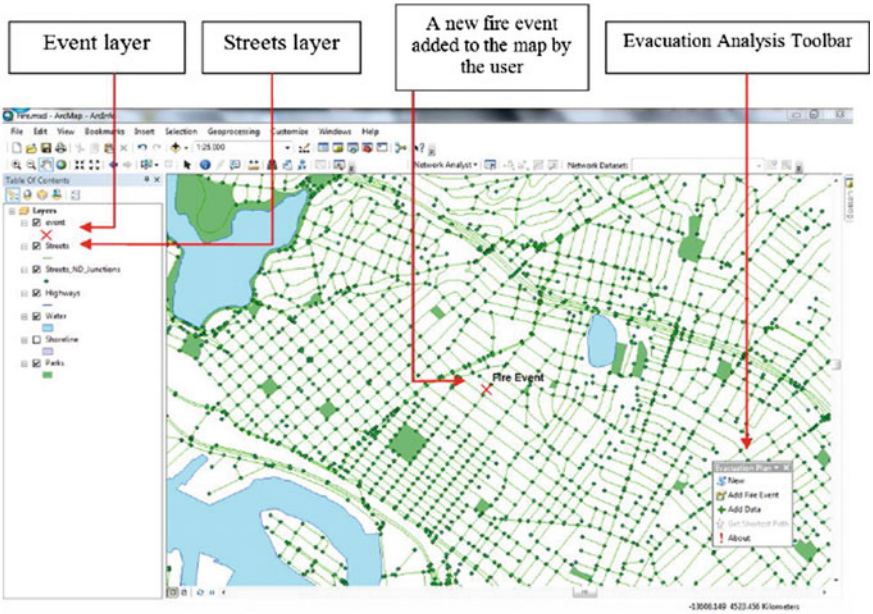


Fig. 4 An ArcMap document with an added streets layer and fire event

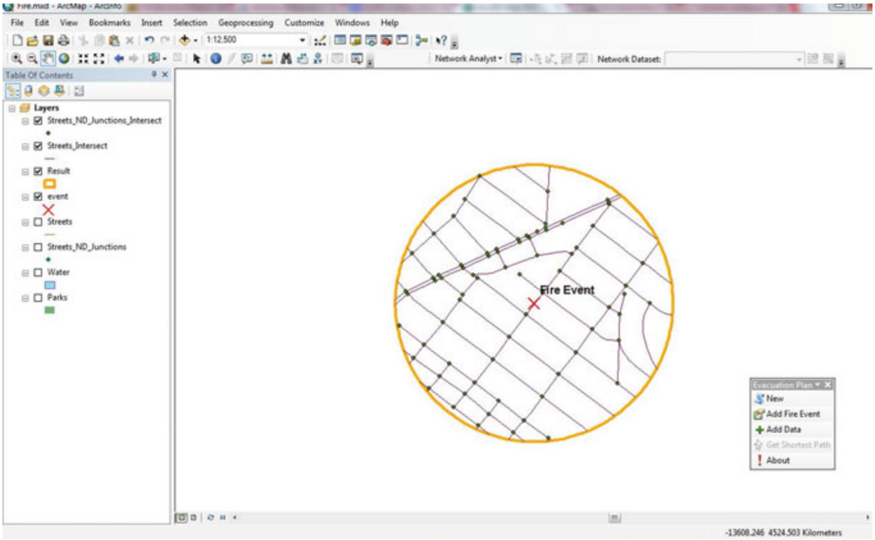


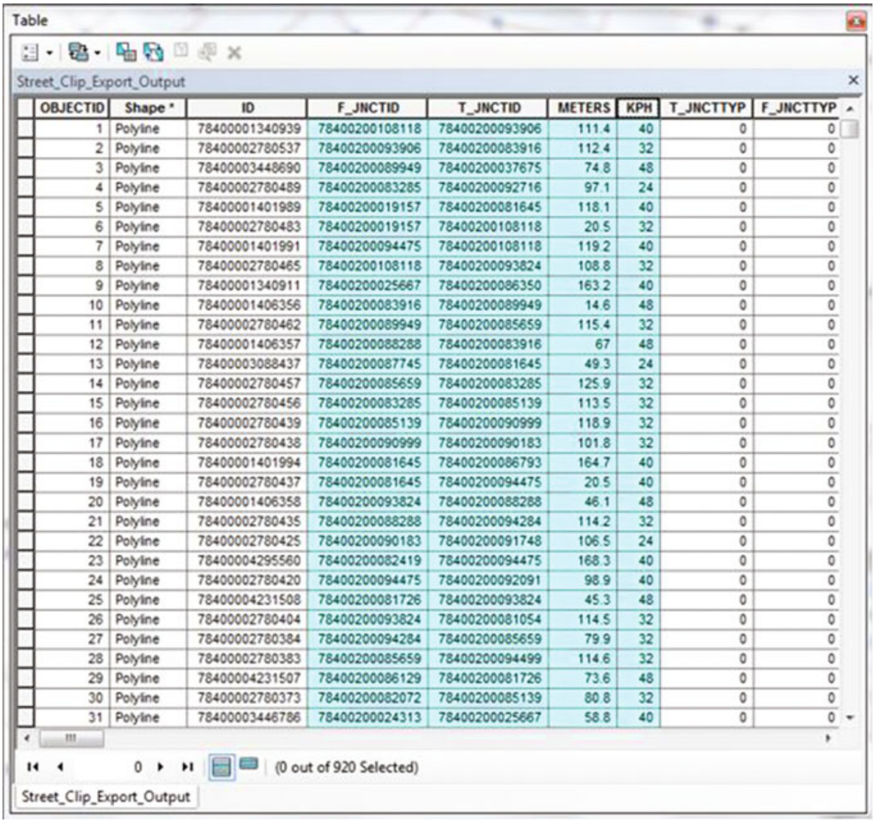
Fig. 5 Intersect layer after overlay analysis

An overlay analysis is performed in order to save the intersection between the streets layer and streets inside the buffer layer. A map with the displayed output of the overlay-analysis tool is shown in Fig. 5.

4.2 G-BEP Evacuation-Path Generation

A network dataset is needed as an input to G-BEP (streets map/layer). The network dataset [25] stores symbology for junctions, edges, and other needed attributes such as distance and speed (to calculate the travel time). The graph (sparse matrix) is constructed with these four attributes (shown in Fig. 6). To run any shortest-path algorithm, a source and a destination should be predefined, except in the case of using Bellman–Ford algorithm, in which no destinations are required. Thus, it computes the shortest paths to all other nodes in the graph, which would be meaningless and time-consuming in our case. To solve that problem, an algorithm for identifying all of the safe nodes outside the buffer is called up (as explained in Sect. 3), and all safe destinations are stored inside an array. Running both Dijkstra's and Bellman–Ford algorithms produces same results. At this point, the work is saved for historical records.

The best three alternatives from a centric source node after obtaining the corresponding Object_IDs are drawn on the shapefile being read into MATLAB® as shown in Fig. 7.



OBJECTID	Shape *	ID	F_JNCTID	T_JNCTID	METERS	KPH	T_JNCTTYP	F_JNCTTYP
1	Polyline	78400001340939	78400200108118	78400200093906	111.4	40	0	0
2	Polyline	78400002780537	78400200093906	78400200083916	112.4	32	0	0
3	Polyline	78400003448690	78400200089949	78400200037675	74.8	48	0	0
4	Polyline	78400002780489	78400200083285	78400200092716	97.1	24	0	0
5	Polyline	78400001401989	78400200019157	78400200081645	118.1	40	0	0
6	Polyline	78400002780483	78400200019157	78400200108118	20.5	32	0	0
7	Polyline	78400001401991	78400200094475	78400200108118	119.2	40	0	0
8	Polyline	78400002780465	78400200108118	78400200093824	108.8	32	0	0
9	Polyline	78400001340911	78400200025667	78400200086350	163.2	40	0	0
10	Polyline	78400001406356	78400200083916	78400200089949	14.6	48	0	0
11	Polyline	78400002780462	78400200089949	78400200085659	115.4	32	0	0
12	Polyline	78400001406357	78400200088288	78400200083916	67	48	0	0
13	Polyline	78400003088437	78400200087745	78400200081645	49.3	24	0	0
14	Polyline	78400002780457	78400200085659	78400200083285	125.9	32	0	0
15	Polyline	78400002780456	78400200083285	78400200085139	113.5	32	0	0
16	Polyline	78400002780439	78400200085139	78400200090999	118.9	32	0	0
17	Polyline	78400002780438	78400200090999	78400200090183	101.8	32	0	0
18	Polyline	78400001401994	78400200081645	78400200086793	164.7	40	0	0
19	Polyline	78400002780437	78400200081645	78400200094475	20.5	40	0	0
20	Polyline	78400001406358	78400200093824	78400200088288	46.1	48	0	0
21	Polyline	78400002780435	78400200088288	78400200094284	114.2	32	0	0
22	Polyline	78400002780425	78400200090183	78400200091748	106.5	24	0	0
23	Polyline	78400004295560	78400200082419	78400200094475	168.3	40	0	0
24	Polyline	78400002780420	78400200094475	78400200092091	98.9	40	0	0
25	Polyline	78400004231508	78400200081726	78400200093824	45.3	48	0	0
26	Polyline	78400002780404	78400200093824	78400200081054	114.5	32	0	0
27	Polyline	78400002780384	78400200094284	78400200085659	79.9	32	0	0
28	Polyline	78400002780383	78400200085659	78400200094499	114.6	32	0	0
29	Polyline	78400004231507	78400200086129	78400200081726	73.6	48	0	0
30	Polyline	78400002780373	78400200082072	78400200085139	80.8	32	0	0
31	Polyline	78400003446786	78400200024313	78400200025667	58.8	40	0	0

Fig. 6 Needed attributes to build the graph

To evaluate how the prototype responds to updates from the dynamic data, the weights of some edges are updated, and the shortest-path algorithm is run again to generate new results. In Fig. 8, it is assumed that two road segments (marked with X) have one of the parameters and thus must be blocked and ignored by the algorithm. Their speed value is set to zero, and thus travel time will equal to infinity. Because the shortest-path algorithm only considers edges with minimal weights, these two edges are ignored, and alternative new paths are generated. Note that paths from the same source node before updating the weights are shown in Fig. 7.

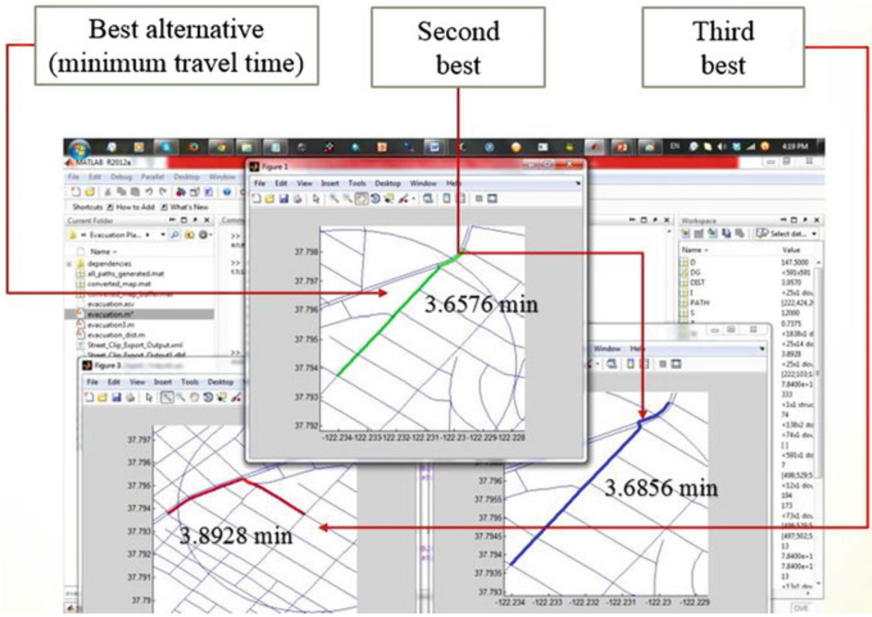


Fig. 7 The best three alternatives from one single source

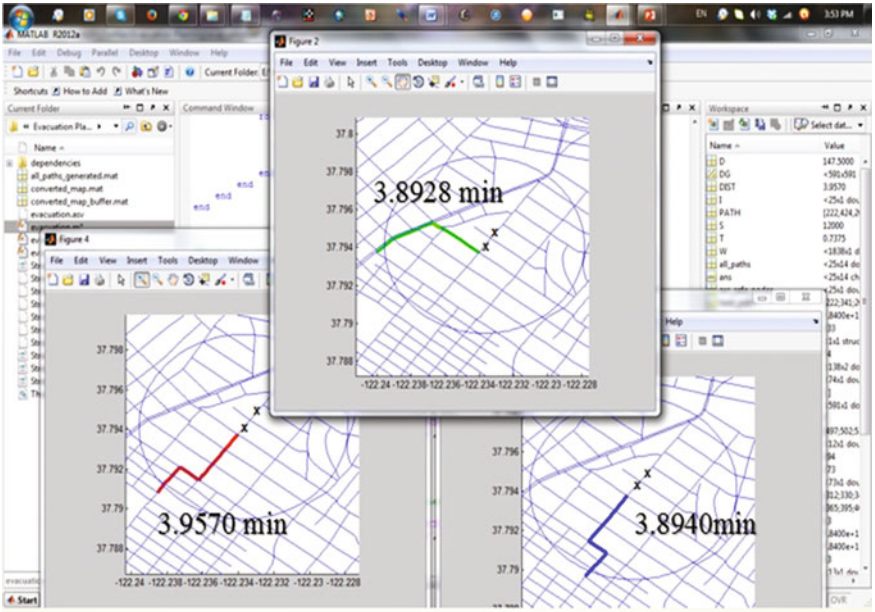


Fig. 8 The best three alternatives after updating the weights

5 Conclusion

This research presents a decision-making framework on the ArcGIS® platform that combines spatial-analysis and graph-theory algorithms with the three presented emergency models. Obtaining alternatives gives the decision-maker more options during a crisis in order to evacuate as many people as possible within an optimum time. The main contributions of this research are the integration of (1) three distinct emergency danger–prediction models incorporating features of the ArcGIS® Spatial Analyst; (2) the addition of a new toolbar in ArcGIS® called Evacuation-Analysis Tool; (3) the proposition of a framework that integrates GIS capabilities, emergency models, and shortest-paths algorithms and considers parameters based on dynamic road conditions, which contribute to the selection of best alternative paths; and (4) identification of the nearest safe destinations outside the buffer zone, not just one destination, but all of the best alternative routes based on graph-theory algorithms (multiple algorithms can be applied, and any number of alternatives can be displayed because all paths are saved).

The findings of this study have a number of important implications for future practice and policy. It is recommended to attach a live network of webcams and streaming-video cameras to the streets and to obtain online data updates through GPX (GPS-Exchange Format) files sent from vehicles or electronic devices at the emergency location, thus producing an evacuation path that depends on the evacuees' status, the weather variables, and the road conditions such as crowd and traffic congestion. Using real-time data would be applicable because all data would be dynamically read through streaming channels.

References

1. Ebada R, Mesbah S, Kosba E, Mahar K (2012) A GIS-based DSS for evacuation planning. ICCTA. Alexandria, Egypt, pp 35–40
2. Cova TJ (1999) GIS in emergency management. In: Longley PA, Goodchild MF, Maguire DJ, Rhind DW (eds) Geographical information systems: principles, techniques, applications, and management. Wiley, New York, pp 845–858, Published
3. Drabek TE, Hoetmer GJ (1991) Emergency management: principles and practice for local government. International City Management Association, Washington DC
4. ESRI (2008) Geographic information systems providing the platform for comprehensive emergency management. ESRI, New York, Published
5. ESRI (2000) Challenges for GIS in emergency preparedness and response. ESRI, New York, Published
6. Bo Y, Yong-gang W, Cheng W (2009) A multi-agent and GIS based Simulation for emergency evacuation in park and public square. Huazhong University of Science and Technology, Wuhan, China
7. Tao C, Hong-yong Y, Rui Y, Jianguo C (2008) Integration of GIS and computational models for emergency management. Center for Public Safety Research, Department of Engineering Physics, Tsinghua University, Beijing, P.R.China

8. Li Y, Yao L, Hu H (2010) Development and application of GIS-based multi-model system for risk assessment of geological hazards. Institute of Geology China Earthquake Administration, Beijing, China
9. Song W, Zhu L, Li Q, Liu Y, Dong Y (2009) Evacuation model and application for emergency events. Tianjin Polytechnical University
10. Jian-guo W, Jian-long Z (2009) Research and development on the expressway emergency response system based on GIS. School of Traffic and Transportation Engineering, Changsha University of Science& Technology, Changsha, China
11. Zhao H, Mao H (2009) Highway emergency response system based on GIS-T. School of Automobile Engineering, Harbin Institute of Technology, Weihai, China
12. Hormdee D, Kanarkard W, Taweepworadej W (2006) Risk management for chemical emergency system based on GIS and Decision Support System (DSS). Department of Computer Engineering, Khon Kaen University, Thailand
13. Zhichong Z, Yaowu W (2009) Framework of spatial decision support system for large-scale public building evacuation. Harbin Institute of Technology, Harbin, China
14. Ling A, Li X, Fan W, An N, Zhan J, Li L, Sha Y (2009) Blue arrow: a web-based spatially-enabled decision support system for emergency evacuation planning. School of Information Science and Engineering, Lanzhou University, Lanzhou, China
15. Zhu L, Song W, Li Q (2009) Construction of emergency decision system based on GIS. Tianjin Polytechnical University, Tianjin, China
16. Wenjing M, Yingzhuo X, Hui X (2009) The optimal path algorithm for emergency rescue for drilling accidents. Institute of Computer Technology, Xi'an Shiyou University, Shannxi, China
17. Shashikiran V, Kumar TS, Kumar NS, Venkateswaran V, Balaji S (2011) Dynamic road traffic management based on Krushkal's algorithm. Department Of Computer Science, Sri Venkateswara College of Engineering, Chennai, India
18. Yuanzhe X, Zixun W, Qingqing Z, Zhiyong H (2012) The application of Dijkstra's algorithm in the intelligent fire evacuation system. Polytechnic Engineering, QiongZhou University, Sanya, China
19. Pettie W (2002) A new approach to all-pairs shortest paths on real-weighted graphs. Theoret Comput Sci 312(1):47–74
20. Banks JH (1991) Two capacity phenomenon at freeway bottlenecks: a basis for ramp metering? Transp Res Rec 1320:83–90
21. Rothermel RC (1972) A Matrematical model for predicting fire spread in wildland fuels. Intermountain Forest And Range Experiment Station, Forest Service, U.S. Department of Agriculture, Ogden, Utah 84401, USA
22. Willis C, Wilgen BV, Tolhurst K, Everson C, D'Abreton P, Pero L, Fleming G (2001) The development of a national fire danger rating system for South Africa. Department of Water Affairs and Forestry Pretoria, Pretoria
23. Carvill Hurrican Index Whitepaper, http://www.cmegroup.com/trading/weather/files/Hurricane-CHIWhitepaper.pdf. Last accessed 25 Aug 2014
24. Saradjian MR, Akhoondzadeh M (2011) Prediction of the date, magnitude and affected area of impending strong earthquakes using integration of multi precursors earthquake parameters. Remote Sensing Division, Surveying and Geomatics Engineering Department, University College of Engineering, University of Tehran, Iran, Copernicus Publications on behalf of the European Geosciences Union, doi:10.5194/nhess-11-1109-2011
25. ESRI (2004) Network dataset defined. Available: http://help.arcgis.com/en/arcgisdesktop/10.0/help./index.html#//004700000007000000. Last accessed 25 Aug 2014