Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231– 6345 (Online)
An Open Access, Online International Journal Available at www.cibtech.org/sp.ed/jls/2015/03/jls.htm
2015 Vol. 5 (S3), pp. 597-608/Jafari et al.
Research Article

PRESENTATION OF AN OPTIMAL LOCATION METHOD FOR EMERGENCY PATH FINDING

Mohamad Jafari, *Marjan Babaei Saeideh, Khorasani Maryam and Talabeygi Farzane
Department of Industrial Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran
**Author for Correspondence*

ABSTRACT
Crisis may occur at any time and, if not properly managed, may inflict huge losses. In a road transportation network, good communication is particularly important in emergency cases. An efficient road network can reduce the impacts of accidents. But a network which cannot provide good access is not reliable and may give rise to broader damages. In this paper, we provide a model based on broad field investigations to determine the importance of transportation arcs. The results obtained from using the proposed model indicated that this method cannot be used individually for determining the importance of project selection and enhancing network arcs in crisis times. In other words, the use of proposed model is not sufficient to determine the importance of transportation network components in crisis times. In this model, importance is generally given to the arcs which reduce the entire network access more than other arcs do. In the proposed method, as problem solving was dependent on road network characteristics, we had to use the software which supported these characteristics in order to facilitate problem solving. For this purpose, we used powerful software of Matlab and defined and solved path finding problem by creating macros.

Keywords: *Emergency Path Finding*

INTRODUCTION
With the expanded urbanization and the increased urban population, one of the most important challenges is finding a solution to reduce the length and duration of urban transportation. In doing so, location information systems play a significant role as they are the means of managing and analyzing the information on current activities.

Among the important capabilities of these systems is network analysis including computation of the shortest path. In the employment of location information systems for path finding, particularly finding paths for firefighting vehicles, ambulances, and the like, special attention has been paid to methods in which the best path is selected based on dynamic travel time. Solving such problems requires that network arc weight be equal to arc travel time.

Computation of the time of each arc is very complicated as it is influenced by many factors such as arc length, existing traffic, arc width, vehicle type, and road coverage type. Since traffic is unpredictable and is in continuous change, a simultaneous navigation system is needed to provide proper answers to users in order to control traffic changes during a day and report any increase and decrease in traffic so that users can make proper reaction when facing a new situation. In such problems, due to the change of arc travel time during the path, one cannot determine the best path using the shortest static route algorithms. To do so, one has to use dynamic algorithms (Alivand and Malek, 2008; Chabini, 1991; Huan and Zhan, 2007).

Since 1959, when Dijkstra presented an algorithm to find the shortest path, there has been an ever-increasing enthusiasm for finding efficient algorithms and many algorithms have been presented (Pallotino *et al.*, 2006; Scutella, 1998). Also, some works have been done in order to find the best path in dynamic transportation networks (Huang, 2007; Ahuja *et al.*, 2002; Djidjev *et al.*, 2000). Cook (1996) was the first to carry out a considerable research to find the best path in time dependent networks in which arc weight changes in a predictable manner. In doing so, he used a time discrete model to solve the problem (Cook and Halsley, 1966). Thereafter, Chabini (1998) classified the problems concerning best dynamic path and proposed two methods based on whether the time is continuous or discrete. Therefore, the division of discrete time into distinct periods with fixed costs was introduced as a fundamental solution to such problems. With the dynamism of transportation network, most research on shortest path algorithms have dealt with static

networks where topology and cost of routing is fixed (Wu, 2006). With the limitations in the capacity of past computer systems, dealing with dynamic network problem has faced much difficulty.
The problem of finding the best path in dynamic transportation networks is called time-dependent shortest path (TDSP) (Dean, 2004). Based on the characteristics of dynamic transportation networks and continuous traffic changes, there are two algorithms for finding the best path in dynamic transportation network (Dean, 2004). One of these methods is re-optimization, in which the best path is computed regularly due to continuous changes in network data.

Another method is re-processing which is one of the most important methods for overcoming the problems in the analysis of the best path in dynamic networks (Wu, 2006; Dean, 2004). In this method, the best path is found and provided to users based on current traffic conditions. When the user is moving in the specified path, if a change occurs in traffic conditions of the path or a part of it, the best path is re-analyzed from the new location toward the destination of user. If a better path is found, the new path is notified to user as the best path. Thus the optimal path is re-computed in order to match the new conditions.

Murchland (1970) conducted a series of studies on path re-processing methods (Murchland, 1970). Most researches on path re-processing method have focused on computation of time complexity of algorithm. So far, many methods have been presented to reduce route computation time in traffic reduction and increase situations (Hung, 2007; Djidjev, 2000; Dean, 2004; Caimi *et al.*, 2012).

Path finding algorithms are divided into two main categories of matrix algorithms and tree structure algorithms (Preggel, 1999). Matrix algorithms find the shortest distance between all pairs in the network. These algorithms consider the network as a matrix.

Tree structure algorithms find the shortest path from origin to other points. These algorithms generate a tree of shortest paths with branches stemming from the origin.

Among tree structure algorithms are Djidjev's algorithm (Coreman *et al.*, 2001) and Belman Ford's algorithm (Detcher and Pearl, 1985). Among matrix algorithms we can mention Floyd-Varshal *et al.*, (1999), Chen *et al.*, (2013); and Yang and Tang (1999) analyzed network performance in critical situations based on certain criteria such as travel time confidence, network connection and network capacity.

Najima and Sugito used simulation method to estimate failure probability in network components in the time of earthquake, and investigated network performance by evaluating traffic changes. Taylor and D'Este (2004), Pellegrini *et al.*, (2004), Tornquist (2012), Qiang and Nagurney (2008), Taylor and D'Este (2004), Qiang (2003) and Rodriguez (2012) evaluated network performance by providing a performance index based on the concept of access and assessing the changes in a number of parameters influencing network performance such as traffic, time, distance and cost of travel.

Among them, Qiang Sohn and Nagurney prioritized risk components of the network. Brabhahran introduced road components prioritization factors and used a scoring method to prioritize network risk components.

In this paper, we pursue two objectives. The first objectives are to location emergency stations; the second is to find the optimal path. In the following paragraphs, we first deal with model design and then investigate the results.

Proposed Model

In this study, we first specify the most important path finding factors in crisis conditions based on interview with experts and specialists. Then, we determine the amount of importance of each factor. We regard the factors with considerable importance weight as feasibility study factors and exclude those arcs which we found unqualified based on these criteria.

Other criteria are considered as risk factors. By defining a numerical index for each factor and defining a linear relationship between them based on the specified importance weights, we attribute an index to each arc as risk index, which will be the basis of path finding.

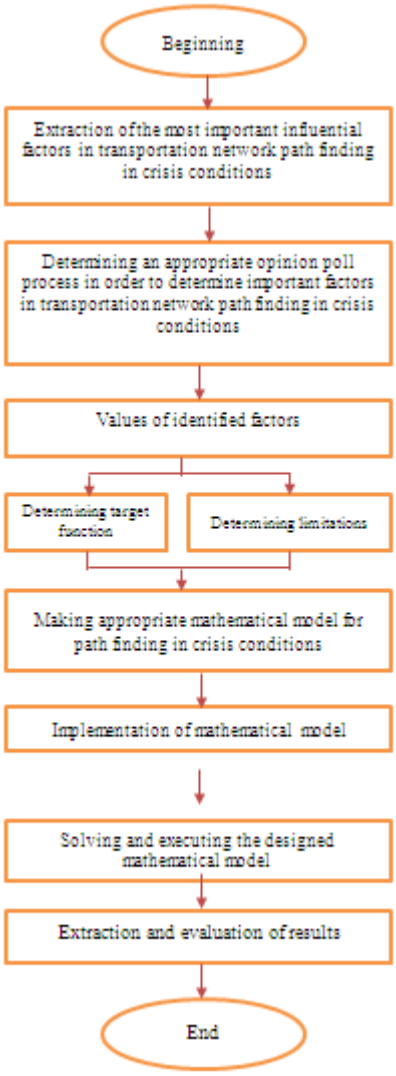

Figure 1: Design Process

Hypotheses

• The amount of transportation demand in the analysis period is constant.
• Cost criterion is the entire travel time of users; walking time is excluded.
• Capacity of sitting in emergency station is 1.
• Individuals reach the station in a random manner and with a steady distribution. Therefore, waiting time equals half of time distance.
• Transportation stations conform to street network nodes.
• Street network is fully continuous, i.e. both street nodes are connected to each other by a network path.

Definition of Network and Query Space

One of the important steps in solving network path finding problem is to determine a space in which optimal path is queried. In this research, we determined query space using Zhao's approach (Zhao, 2006; Zhao and Zheng, 2008). The basic idea of this approach is the selection of main nodes on the initial path and then the selection of nodes which are connected to the nodes of main path with one, two, three and more arcs. The use of these nodes generates new paths with the centrality of main path, which play a significant role in path optimization. Emergency transportation network ($T=T(t_{ij})$) is the matrix of arcs which specifies each arc on an emergency transportation path.

Introduction of Problem Limitations

In discrete optimization, due to difficulty of problem and for preventing the expansion of problem dimensions, less decision variables are selected. On the other hand, limitations must meet the needs of problem as far as possible. In this research the following limitations have been applied:

• Continuity of transportation choice (e.g. emergency) in a path

• Maximum travel time in a vehicle (or the length of path) for emergency transportation paths. In long paths, tiredness of drivers makes some difficulties. In short paths, there are emergency planning problems.

• Fleet dimensions in each line: The number of emergency transformation fleet affects functional cost. Network optimization problem must be able to determine the best path and time period by minimizing user time and consider budget limitations.

• Minimum and maximum time period in a given path: In order to avoid long waiting times, time period should be neither too short nor too long. Time period is also associated with vehicle fullness coefficient, which is related to the number of passengers inside vehicle.

• Directness of path: Directness of path must be more than minimal amount. In emergency transportation, the possibility of using shortest origin-destination path is not necessarily available.

Optimal Location

There are different indexes for measuring spatial correlation. In this study we used the indexes of Moran and Getis-Ord Gi to investigate about spatial distribution of location quality values. Moran's statistic is one of the best indexes for identification of clustering. This statistic specifies whether adjacent areas have similar or non-similar values. The value of Moran varies between 1 and -1. The values closes to 1 indicate that the areas with similar values have a cluster pattern; the values close to -1 indicate that the areas with non-similar values are close to each other; the value of zero indicates a stochastic pattern.

Moran's index is defined as follows:

$$I = \frac{n \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (x_i - \bar{x})(x_j - \bar{x})}{(\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}) \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Where n is the number of samples, x_i is the value of variable in area i, x_j is the value of variable in area j, $\bar{x}$ is the average of variable in all areas, and w_{ij} is the weight used to compare two areas of i and j.

Getis and Ord (1992) presented Gi statistic to identify local clusters in which the areas with similar values (higher or lower than average) are close to each other as follows: (HotSpot analysis)

$$G_i^* = \frac{\sum_{j=1}^{n} w_{ij} x_j - \bar{x} \sum_{j=1}^{n} w_{ij}}{S \sqrt{\frac{\left[n \sum_{j=1}^{n} {w_{ij}}^2 \left(\sum_{j=1}^{n} w_{ij}\right)^2\right]}{n-1}}}$$

Where Gi is the value of Getis index for *i*m polygon, S is standard deviation of the values of areas. Other parameters are according to equation 9. The value of Gi for each area indicates if that area is close to other areas with higher (for positive values of Gi) or lower (for negative values of Gi) than the average.

Closest Neighbor Average Index

Closest neighbor average index is based on measurement of distance of each user to the closest neighbor. This index is used to determine convergence and divergence of different kinds of applications. This analysis attempts to determine whether or not the distribution of points is random and how dispersion pattern is (Camarero *et al.*, 2000). In this method, closest neighbor index is measured based on the average distance of each user to the closest neighbor. Closest neighbor index is expressed as the ratio of average observed distance to the expected distance. In this method, the expected distance is obtained by analyzing Z quantity. If this value is between 1.96 and -1.96, there is a significant difference between distribution and there is no random distribution. Otherwise, distribution would be accumulative or steady.

Closest neighbor average index is obtained from the following equation:

$$ANN = \frac{\bar{D}_O}{\bar{D}_E}$$

Where $\bar{D}_O$ is the average distance between each index and the closest neighbor, which is obtained from the following equation:

$$\bar{D}_O = \frac{\sum_{i=1}^{n} d_i}{n}$$

The average expected distance for the index of a random pattern:

$$\bar{D}_E = \frac{0.5}{\sqrt{n/A}}$$

In the previous equation, $\bar{D}_E$ is the distance between index i and its closest neighbor, n is total number of indexes, and A is the entire areas under study

Location Patterns

Below are location patterns used for locating urban emergency stations:

• Maximum Coverage Location Pattern (MCLP)
• P-Median Pattern (minimum distance)
• P-Center pattern (minimization of the highest distances between demand points and closest facilities)
• Other patterns

P-Media and P-Center Patterns

P-Median and P-Center patterns are based on the type of target function. In these patterns, the reduction of relief time is of particular importance. However, these patterns are not appropriate for making models whose objective is to find optimal arrangement of relief station networks.

MCLP Patterns

MCLP patterns are more appropriate for finding optimal location of emergency relief stations because their target function is maximization of demand coverage amount.

Computation of Criteria Weight

In the new path finding model, we investigate the effective parameters in the paths ending on temporary accommodations in crisis times and prioritize the parameters of the path by forming paired comparison matrixes. To compute the relative weight of the criteria, we first have to compute the normalized comparisons matrix. Thus, if comparison matrix (A) for parameter (n) is according to equation (1), we use equation (2) for normalized comparisons matrix and finally obtain the matrix of equation (3).

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \dots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}$$

$$\acute{a}_{ij} = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}}$$

$$\acute{A} = \begin{bmatrix} \acute{a}_{11} & \dots & \acute{a}_{1n} \\ \vdots & \dots & \vdots \\ \acute{a}_{n1} & \dots & \acute{a}_{nn} \end{bmatrix}$$

We compute the average number of each line of matrix, which is the relative weight of decision parameters.

$$W_i = \sum_{j=1}^{n} \acute{a}_{ij}$$

Where Wi is the relative weight of parameter (i).

Finally, we multiply the relative weight of each component (parameter) by the weight of higher components (parameters) to compute the final weight of parameter. Below is the final model for computing the value of each choice and selecting better choice in order to manage transportation in crisis times:

Alternative Point=$\sum_{i=1}^{n} U_i \times F_i$

Where Fi is the normalized amount of the intended parameters.

Based on the results obtained from interview with experts, we extracted the relative weight of the parameters, as shown in Figure 2.

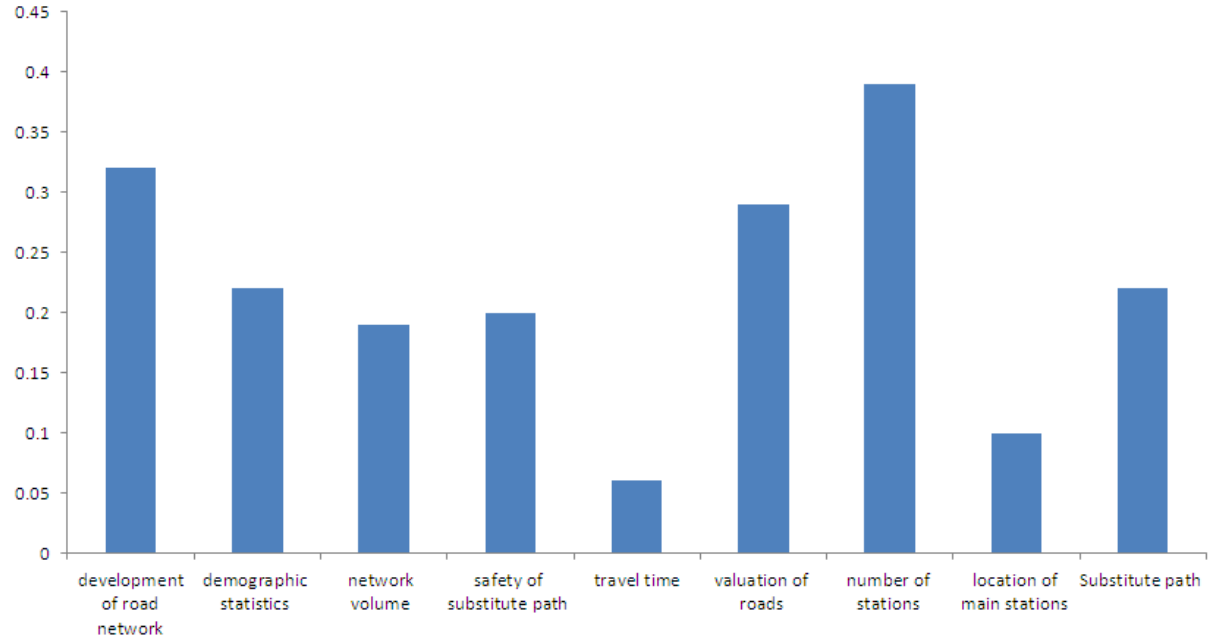

Figure 2: Coefficients of the main parameters

Optimal Path Finding

In this research, we used Dijkstra's algorithm with weighting arcs to determine the shortest path between origin and destination (traffic areas). This algorithm is a graph survey algorithm which analyzes shortest path problem for different graphs and finally provides the shortest path from origin to destination by developing a tree. Below is the procedure of this algorithm:

1. Selection of origin point
2. Determination of S collection including heads of the intended graph
3. Placing origin point with zero index inside S
4. For the heads outside S, we apply an index amounting to arc length + previous head index. If the head outside the collection has index, the reduction of amount is permissible only.
5. From the heads outside S, we select the head with least index and place it in S.
6. Going to fourth step until reaching destination node.

Determination of the weights of parameters and sub-parameters

$$y_{ij}^{sL} = n_{ij}^{sL}$$

$$K_{ij}^{sL} = q_{ij} \times y_{ij}^{sL}$$

$$A_i^{sL} = \sum_j K_{ij}^{sL}$$

$$B^{SL} = \sum_i P_i \times A_i^{sL}$$

X_{ij}^{sL} : Observation of sub-parameter (j) of parameter (i) in arc (L) of axis (S)

S: axis index

L: arc index

I: parameter index

J: sub-parameter index

N_{ij}^{sL} : The number of values of sub-parameter (j) of parameter (i) in arc (L) of axis (S)

n_{ij}^{sL} : Standardized values of

$\mathrm{N}_{\mathrm{ij}}^{\mathrm{sL}}$: Amended values of for conformity of collected information and sub-parameters

P_1: parameter weight
q_{ij}: parameter weight
K_{ij}^{sL}: Computed values of sub-parameters separated into arcs (item weight)
A_i^{sL}: Computed values of parameters separated into arcs
B^{SL}: Computed values of parameter combination separated into network arcs

Dijkstra's Path Finding Principles

In shortest path methods, labeling method is very important (Gallo & Pallottino, 1988; Ahuja *et al.*, 1993). The output of this method is a tree network which leaves the origin point *S* and goes to a collection of other point in the network. This network is developed repeatedly until the shortest path between *S* and *i* is obtained and execution comes to an end. In labeling process, three types of information are kept for each point of the network:

• Distance label d(i) which represents the shortest distance between *S* and *i* during repetition of model. Once the model finishes, d(i) represents the shortest distance between *S* and *i*.
• Parent point label p(i) which represents parent point of *i* in the developed tree network.
• Point status label S(i) which may be without label or have temporary or permanent label. If a point is not surveyed during the repetition, it will be without label and its distance label is set at infinite positive. When the shortest path until point *i* is definitely the shortest possible path, the label of point *i* is set at permanent labeled status.

When the path is expected to improve in the next repetitions, label of point *i* is set at temporary labeled status. In the primary Dijkstra's method, each point can only have one of two statuses of temporary labeled and permanently labeled. In each repetition, therefore, the model selected the point with temporary labeled status or minimum distance label in relation to the next point (Dijkstra, 1959; Ahuja *et al.*, 1993).

The procedure of labeling point *i* is as follows:

Procedure ScanningOperation(i)
Begin
for all successor nodes of i do
if d(i)+1(ij)<d(j) then
begin
d(j)=d(i)+1(Ij);
p(j)=i;
s(j)=labeled;
end
s(i)=permanently labeled;
end

The general idea of the proposed algorithm for finding optimal path of emergency evacuation problem is based on Dijkstra's approach. In this method, we first compute cost (C) for all arcs and attribute infinite passage cost (d) to all graph points. Then, for each arc between two points of *i* and *j*, which is located between start and end points, we check if d(i)+c(ij)<d(j)? If d(j)=d(i)+c(Ij), we add the arc between two points of (i) and (j) to the end of optimal path. Otherwise, we stop searching this path at the location of that arc.

In the proposed method, apart from the requirement of d(i)+c(ij)<d(j) for arc (ij), we apply three requirements.

The first requirement is d(i)+c(ij)<min{d(j)d(e)}, where *e* is destination point. This prevents the continuation of path search with higher cost. If this method is applied for one origin point and several destination points at the same time, d(e)=max{d(ei)} must be met in the above equation. According to the second requirement, the path can only cross the points which have not been passed before.

In other words, before arc cost is tested in the above requirement, it is checked that the new point of arc does not exist in the current points of the path.

The third requirement refers to the order of searching the arcs relating to different paths. Based on this requirement, searching priority is always given to the path with the lowest cost.

RESULTS AND DISCUSSION

Results

In order to investigate the efficiency of the proposed method, we implemented the algorithm in MATLAB and tested it on simulated data. For the purpose of simulation, we created a graph of 20,000 points with random situation (0<xy<1) and performed triangulation. The sides of triangles represent graph arcs. In this example, the number of graph arcs is 1,000. Next, we attributed a weight in the range of 1-10 to each arc. To ensure that the algorithm works property, we searched the path between two points of the network independently on a non-directional network by replacing origin and destination points and observed that the paths fully conform to each other. After repeating the test for several modes with different origins and destinations, we ensured that the algorithm works properly.

In the second test, we examined the efficiency of general and local path between the two points. To determine the optimal local path, we searched 5,000 arcs and obtained a path consisting of 39 points with the cost of 107.14. To determine the optimal general path, we searched 1,897 arcs and obtained a path with 32 arcs with cost of 104.06.

As you can see, the application of almost similar computational time to general and local paths gives different final results. This indicates the necessity of searching general optimal path compared with optimal local path. To ensure that the algorithm works properly in general and local optimal paths, we performed the test on 100 origin and destination points. The results indicated that optimal general path improved about 30% compared with local path. In 55% of the cases, no improvement was seen as the paths were the same. On the other hand, computational cost for finding the optimal general path increased by around 90% compared with local path. In general, the results indicated that the algorithm was more efficient for searching optimal general path compared to local path. With increase of computational cost by 9%, we can determine optimal general paths instead of optimal local paths. It should be noted that these results also indicate the complexity of simulated network. In the third test, we examined the efficiency of algorithm in searching optimal paths between an origin and several destinations at the same time. To determine an optimal path, we searched 10,000 arcs. To determine five optimal paths at the same time, we searched 2,107 arcs. In other words, with the increase of computational cost by 27%, five optimal general paths were found instead of one optimal path. This indicates that in emergency evacuation operation for each building, several optimal paths to neighboring safe areas must be proposed so that other paths can be used when one path is blocked. Therefore, we found the algorithm to have a good efficiency. For each simulated network pair, the algorithm searched the optimal path of different points for ten times based on the characteristics of Table 1. The average processing time and percentage of arcs have been recorded in Table 1. According to the results, around 22% of network arcs have been processed for searching the optimal general path.

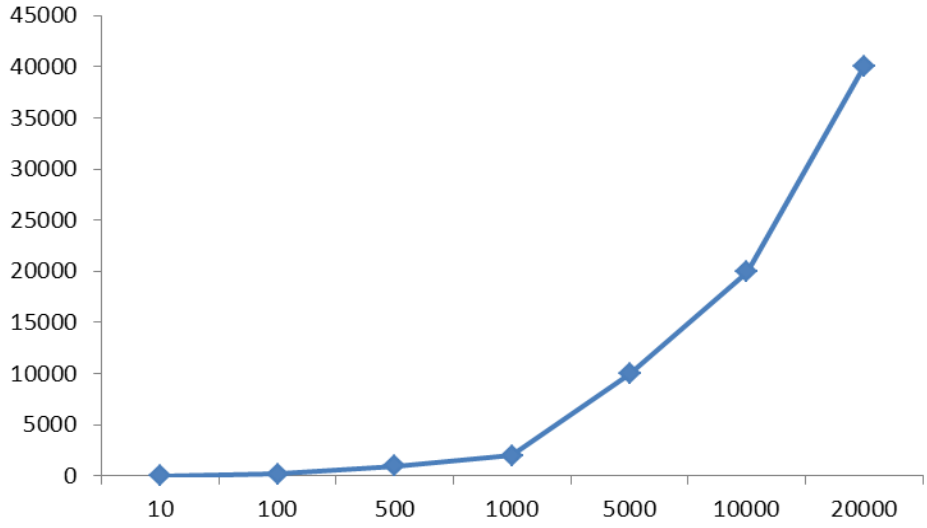

Figure 3: Number of network arcs

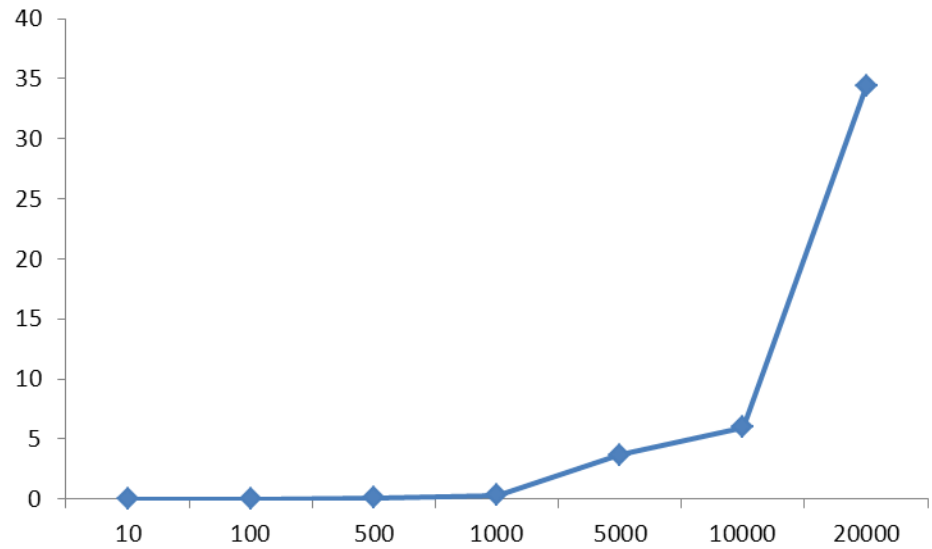

Figure 4: Average processing time (sec)

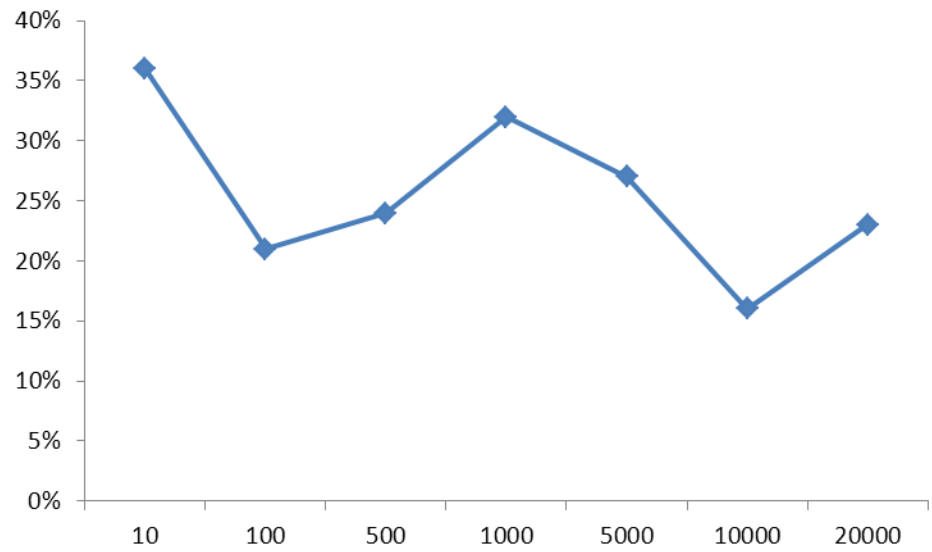

Figure 5: Percentage of searched arcs

In this test, we compared the proposed algorithm with the well-known algorithms in the literature. For this purpose, we selected P-Median, P-Center and MCLP algorithms.

Table 1: The results of comparison between the proposed algorithm with other ones

Proposed Algorithm		**MCLP**		**P-Center**		**P-Median**		**Size**
Time	Cost	Time	Cost	Time	Cost	Time	Cost	
0.12	500.78	0.82	512.89	2.28	524.89	6.2	500.61	50
0.22	829.64	11.98	839.54	27.80	812.98	18.75	827.54	100
0.49	10.35.98	18.65	1384.52	48.51	1068.64	59.45	1.098.84	150

We implemented the proposed algorithm in Matlab. Considering the random mechanism in the selection of neighboring answers, the answers provided by the algorithm for each execution were not similar. The results

provided for the proposed algorithm are the best results obtained from five consecutive executions of the algorithm and the results relating to other algorithms have been directly extracted from the relevant papers. As you can see in the table, the answers provided by the proposed algorithm are close to those of the best available algorithms. In three cases, the proposed algorithm provides the best answer. It should be noted that the selected algorithms have been designed for solving emergency problem and are not flexible enough to solve problems with different limitations. Therefore, considering that the proposed algorithm provides answers comparable to the best available algorithms, it is able to effectively solve scientific problems. Since each algorithm has been executed on a distinct machine with particular speed and characteristics, direct comparison of the time spent for solving the problems is not possible. Generally, the proposed algorithm is able to compete with the advanced algorithms in terms of quality and speed of answering.

Conclusion

Among different services provided in a city, medical centers are of special importance. In the recent years, due to the rapid growth of urban population and the lack of comprehensive planning and management, medical sector has faced serious challenges including improper distribution, lack of optimal locations, and lack of necessary measures for distribution of services in the cities. In this paper, we provided a model based on broad field investigations to determine the importance of arcs. We used the proposed model based on Dijkstra's algorithm and found that this method cannot be used individually for determining the importance of project selection and enhancing network arcs in crisis times. In other words, the use of proposed model is not sufficient to determine the importance of transportation network components in crisis times. In this model, importance is given to arcs which reduce the entire network access more than other arcs do. To determine the shortest travel time between origin and destination (O-D), we used Dijkstra's algorithm with weighting arcs. This algorithm is a graph survey algorithm which analyzes the shortest path problem for different graphs and finally provides the shortest path from an origin to different destinations by creating a tree. In the proposed method, as problem solving was dependent on road network characteristics, we had to use the software which supported these characteristics in order to facilitate problem solving. For this purpose, we used powerful software of Matlab and defined and solved path finding problem by creating macros.

REFERENCES

Ahmed MS and Cook (1979). Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transportation Research Record* **722** 1–9.

Ahuja RK, Orlin JB, Pallottino S and Scutella MG (2003). Dynamic shortest path minimizing travel times and costs. *Networks* **41**(4) 197-205.

Alivand M (2008). Navigation services in dynamic traffic by mobile geospatial information systems. M.Sc thesis K.N. Toosi University of Technology Iran.

Alivand M, Alesheikh AA and Malek MR (2008). New method for finding optimal path in dynamic networks. *World Applied Sciences Journal* **3**(1) 25-33.

Azizi A (2003). Network analysis for urban traffic in a GIS. M.Sc Thesis Kn. Toosi University of Technology.

Brabhaharan P, Fleming MJ and Lynch R (2001). Natural hazard risk management for road networks part I: Risk management strategies. *Transfund New Zealand Research Report,* Number 7-4938-1.

Brabhaharan P and Moynihan S (2002). Natural hazard risk management for road network part II: Implementation strategies. *Transfund New Zealand Research Report* No.2002.

Camps VBG, de Silva PAL and Netto POB (2000). Evacuation transportation planning: A method of identify optimal independent routes Urban Transport and the Environment for the 21st Century 555-564.

Chabini I (1998). Discrete dynamic shortest path problems in transportation application. Complexity and Algorithms with Optimal Run Time Transportation Research Record 1645 170-175.

Chen X (2001). Agent-based simulation of evacuation strategies under different road network structures Texas Center for Geographic Information Science Department of Geography Shouthwest Texas state university.

Chen X, Hu J and Hu X (2007). On the minimum risk-sum path problem. *Lecture Notes in Computer Sciences,* Springer 4614 175-185.

Clark S (2003). Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering* **129**(2) 161-168.

Cooke KL and Halsley E (1966). The shortest route through a network with time-dependent intermodal transit times. *Journal of Mathematical Analysis and Application* **14** 275-323.

Cormen TH, Leiserson CE, Rivest RL and Stein C (2001). *Introduction to Algorithms* (MIT Press and McGraw-Hill) 588-601.

Dean BR (2004). Shortest paths in FIFO time dependent networks: Theory and algorithms. Technical Report Massachusetts Institute of Technology U.S.A.

Dechter R and Pearl J (1985). Generalized best first search strategies and the optimality of A*. *Journal of the ACM* **32**(3) 505-536.

Djidjev H, Pantziou G and Zaroliagis C (2000). Improved algorithms for dynamic shortest paths. *Algorithmica* **28**(4) 367-389.

Dunn CE and Newton D (1992). Optimal routes in GIS and emergency planning applications. *Area* **24(**3) 259-267.

Frigioni D, Matchetti-Spaccamela A and Nanni U (2000). Fully dynamic algorithms for maintaining shortest path trees. *Journal of Algorithma* **34**(2) 251-281.

Caimi GG, Fuchsberger M, Laumanns M and Lüthi M (2012). A model predictive control approach for descrete-time rescheduling in complex central railway station approach. *Computers & Operations Research* **39** 2578–2593.

Corman FF, D'Ariano A, Pacciarelli D and Pranzo M (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B* **44** 175–192.

Guhnemann A, Schafer RP, Theissnhusen K and Wagner P (2004). Monitoring traffic and emissions by floating car data. Institute of Transport Studies The University of Sydney and Monash University.

Heil and Walnut (1989). Continuous and discrete wavelet transforms. *Society for Industrial and Applied Mathematics Review* **31**(4) 628-666.

Huang B, Wu Q and Zhan FB (2007). A shortest path algorithm with novel heuristics for dynamic transportation network. *International Journal of Geographical Information Science* **21**(6) 625-644.

Ide T and Kato S (2008). Travel-time prediction using Gaussian Process Regression: A trajectory based approach. IBM Research Tokyo Research Laboratory.

Klunder GA and Post HN (2006). The shortest path problem on large scale real road networks. *Networks* **48**(4) 182-194.

Lusby RM, Larsen J, Ehrgott M and Ryan DM (2013). A set packing inspired method for real-time junction train routing. *Computers & Operations Research* **40**(3) 713–724.

Miller HJ (1999). Measuring space-time accessibility benefits within transportation networks: basic theory and computational methods. *Geographical Analysis* **31** 187-212.

Miller-Hooks E and Yang BY (2005). Updating paths in time-varying networks given arc weight changes. *Transportation Science* **39**(4) 451-464.

Murchland JD (1970). A fixed matrix method for all shortest distances in a directed graph and for the inverse problem. PhD thesis University of Karlsruhe Germany.

Nakata T and Takeuchi JI (2004). Mining traffic data from Probe-Car system for travel time prediction. Internet System Research Laboratories NEC Corporation Japan.

Nojima N and Sugito M (2000). Simulation and evaluation of post earthquake functional performance of transportation network Proc.

Pallottino S and Scutella MG (1998). Shortest path algorithms in transportation models: Classical and innovative aspects. In: *Equilibrium and Advanced Transportation Modeling,* edited by Marcotta P and Nguyen S (Kluwer) 245-281.

Pellegrini P, Marlière G and Rodriguez J (2012). Real time railway traffic management modeling track-circuits. In: *12th Workshop on Algorithmic Approaches for Transportation Modelling Optimization and Systems,* edited by Delling D and Liberti L, Open Access Series in Informatics (OASIcs), Dagstuhl Germany **25** 23–34.

SchlossDagstuhl–Leibniz-ZentrumfuerInformatik, Pellegrini P, Marlière G and Rodriguez J (2013). A mixed-integer linear program for the real-time railway traffic management problem modeling track-circuits. In: *Rail Copenhagen 2013 5th International Seminar on Railway Operations Modeling and Analysis Copenhagen Denmark*.

Preygel A (1999). Path finding: A comparison of algorithms. *Management Science Matthews*.

Thompson E (2003). Integrating PDA GPS and GIS technologies for mobile traffic data acquisition and traffic data analysis. M.Sc Thesis IT University of Goteborg.

TörnquistKrasemann J (2012). Design of an effective algorithm for fast response to re-scheduling of railway traffic during disturbances. *Transportation Research Part C* **20** 62–78.

Vanicek and Wells (1972). The least squares approximation and related topics" Department of Surveying Engineering University of New Brunswick.

Wu JJ, Sun HJ, Gao ZY and Li SB (2007). Effects of route guidance systems on small-world network. *International Journal of Modern Physics C* **18**(8) 1243-1250.

Wu Q (2006). Incremental routing algorithms for dynamic transportation networks. M.Sc thesis Calgary University Canada.