Contents lists available at ScienceDirect

Socio-Economic Planning Sciences

journal homepage: www.elsevier.com/locate/seps

SOCIO-ECONOMIC PLANNING SCIENCES
The Journal of
PUBLIC SECTOR DECISION MAKING

# Debris removal during disaster response: A case for Turkey

Halenur Sahin [a], Bahar Yetis Kara [b,*], Oya Ekin Karasan [b]

[a] *Department of Industrial Engineering, Middle East Technical University, Ankara 06800, Turkey*
[b] *Department of Industrial Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey*

ARTICLE INFO

*Article history:*
Available online 21 October 2015

*Keywords:*
Humanitarian logistics
Debris removal
Mathematical modeling

ABSTRACT

Debris occurs from the ruin and wreckage of structures during a disaster. Proper removal of debris is of great importance because it blocks roads and prohibits emergency aid teams from accessing disaster-affected regions. Poor disaster management, lack of efficiency and delays in debris removal cause disruptions in providing shelter, nutrition, healthcare and communication services to disaster victims, and more importantly, result in loss of lives. Due to the importance of systematic and efficient debris removal from the perspectives of improving disaster victims quality of life and allowing the transportation of emergency relief materials, the focus of this study is on providing emergency relief supplies to disaster-affected regions as soon as possible by unblocking roads through removing the accumulated debris. We develop a mathematical model for the problem that requires long CPU times for large instances. Since it is crucial to act quickly in an emergency case, we also propose a heuristic methodology that solves instances with an average gap of 1% and optimum ratio of 80.83%.

## 1. Introduction and problem definition

Though ultimate avoidance of natural disasters is likely impossible, disaster management operations play a vital role in minimizing the negative effects of disasters and loss of life. The disaster management literature is comprised of studies that focus jointly on preventive strategies for pre-disaster and damage reduction operations for post-disaster periods. The disaster management cycle consists of four phases: preparation, response, recovery and reconstruction. The preparation phase covers precautions that are taken before a disaster occurs to minimize negative outcomes. The response phase starts immediately after the disaster and involves providing emergency services to as many victims as possible as soon as possible. During the recovery phase, the main focus is to restore the disaster area in terms of communication, transportation and infrastructure; and finally, the main objective of the reconstruction phase is to fully rehabilitate the disaster area and normalize the daily lives of disaster victims.

As a result of the destructive effects of disasters, debris, occurs. Proper removal of debris is extremely important for unblocking roads and allowing emergency aid teams to access the disaster area. In this study, we focus on debris removal in the response phase of earthquakes. Complete debris removal may be postponed until the recovery phase, whereas removing debris on routes to critical disaster areas must be done in the response phase. Therefore, effective debris removal is necessary to access disaster victims.

During the response phase, since time is of the utmost significance, it is necessary to determine which areas to access first. It is critical to provide emergency aid as quickly as possible to districts that contain schools, hospitals, potential shelter areas, etc. In order to do so, however it may be necessary to travel on a path that includes blocked roads. In such a case, it is required to unblock these roads by debris removal operations. In this context, we define the *Debris Removal Problem in the Response Phase (DRP)* as visiting prespecified critical disaster-affected districts as quickly as possible by removing debris on blocked roads if and when necessary. Proper distribution of resources will entail timely access to emergency supplies and will help defuse the post-disaster crisis. In accordance with this purpose, the disaster area is assumed to be aggregated into districts. Critical districts and the districts serving as the resource base are determined. The proposed methodology also determines the critical path for the emergency-response vehicle, called RESCUE (Relief Supply Carrier Under Emergency), which transfers relief materials to critical districts. Associated with the critical path, the arcs that require debris removal to resolve blockage and provide access are also identified. It is worth noting that all blocked edges on the critical path must be unblocked, and once an edge is freed from debris, it remains open for later use. Debris removal operations for blocked roads require extra effort,

* Corresponding author.
*E-mail addresses:* halenur.sahin@metu.edu.tr (H. Sahin), bkara@bilkent.edu.tr (B.Y. Kara), karasan@bilkent.edu.tr (O.E. Karasan).

http://dx.doi.org/10.1016/j.seps.2015.10.003
0038-0121/© 2015 Elsevier Ltd. All rights reserved.

which is measured in terms of time in our approach. Traveling more than once on an edge from which the debris has been removed might be advantageous and in determining the critical path for traversing all critical districts, the algorithm will utilize this advantage.

The DRP is an NP-hard problem because the case when there are no blocked edges and all nodes are critical is equivalent to the travelling salesman problem (TSP) [1].

Since there are nodes that must be visited, and since an arc-routing aspect is inherently present, our problem can be defined as a variant of the general routing problem (GRP), which will be detailed in the following section. Different than the GRP, our problem implies that the only reason to traverse an arc is to reach a required node. To the best of the authors' knowledge, this variant of the general routing problem has not yet been defined in the literature.

Organization of the paper is as follows: In Section 2, related literature is visited. Development of mathematical model is in Section 3. In Section 4, heuristic solution methodologies are presented. Section 5 is devoted to computational experiments and their results. Finally, concluding remarks are in Section 6.

## 2. Related literature

The GRP aims to find a minimum-cost vehicle route that starts and ends at the same node and visits the required nodes at least once by passing through the required edges at least once. The required node set is a subset of all nodes and the required edge set is a subset of all edges [2]. The GRP includes both node- and arc-routing aspects, thus node-routing and arc-routing problems arise as special cases. For a comprehensive survey, we investigate both the arc-routing problems (ARP) and the node-routing problems, namely, the vehicle routing problem (VRP). Since one of the key aspects of our problem is debris removal on arcs in order to unblock them, we examine the arc-routing literature in detail.

### 2.1. Arc-routing problems

In ARPs, the aim is to find a minimum-cost vehicle tour that traverses through a specified arc subset, which begins and ends at the same node. The Chinese postman problem (CPP), the rural postman problem (RPP) and the capacitated arc-routing problem (CARP) are primary arc-routing problems. The difference between the GRP and the ARP is that the GRP also considers the node-routing aspect by visiting some nodes on the graph. When the required node set is empty and the purpose is to visit all edges, the GRP reduces to the CPP. On the other hand, if there is a subset of edges that needs to be visited with an empty required node set, then the GRP reduces to the RPP [2,3]. The CPP was first defined by Kwan-Mei Ko in 1962 [4,5] to find a minimum-cost tour that traverses all the arcs of a graph at least once. Waste collection, street sweeping and snow plowing operations, where it is required to pass through all arcs in the graph, are in the application area of the CPP. In 1974, Orloff [6,5] defined the RPP, where the objective is to find a minimum-cost tour that traverses only a subset of arcs, which are called required arcs, at least once. Lenstra and Rinnooy Kan [2] proved that both undirected and directed versions of the RPP are NP-hard. However, if the required edges are all edges of the graph, then the problem becomes a CPP [7]. Street sweeping, snow plowing, garbage collection, mail delivery, school bus routing and meter reading are the most common application areas of the RPP. When a capacity constraint of the vehicle is included, the problem is referred to as the capacitated arc-routing problem, which was first defined by Golden and Wong in 1981 [8]. There are many variations of CARP, and their application areas are also various, including winter gritting, refuse collection, mail delivery, street sweeping operations and police patrols. The aim of the problem is to find a minimum-cost traversal of all arcs such that each arc is serviced without exceeding the capacity of the vehicle.

### 2.2. Node-routing problems

Node-routing problems are special cases of the GRP, of which the VRP is one of the most famous. When there is a subset of nodes required to be visited with an empty required edge set, the GRP reduces to the VRP. Since the general VRP literature is too broad, we only focus on the VRP literature over blocked networks. One of the problems of the shortest-path classification is the Canadian traveller problem (CTP). In their article, Xu et al. refer to the CTP as an abstraction of the online shortest-paths/routing problems [9].

The CTP was first defined by Papadimitriou and Yannakakis and proven to be an NP-hard problem [10]. It is defined for a single source and a single destination and the aim is to find the minimum-cost route from source to destination. The traveller knows the graph structure and edge costs but some edges may be blocked, which the traveller does not know until he/she reaches the adjacent node of this blocked edge [9]. The classic version of the CTP is a stochastic problem and the blocked edges remain blocked forever [11,12]. Note that if all road blockages are known in advance, the optimal travel path can be obtained by applying a shortest-path algorithm from source to destination. However, since the problem has an online nature, the optimal travel strategy cannot be given by the shortest path [11]. In their study, Bar-Noy and Schieber introduced variations of the CTP [11], one of which is the recoverable-CTP, where blocked roads may become open again. There are both stochastic and deterministic versions of the recoverable-CTP. In the stochastic version, each edge has a blockage probability, and in the deterministic version, there is a fixed bound on the total number of potential blockages. In the recoverable-CTP, edges have recovery times. It is assumed that the recovery times of blocked edges incident to the same node are the same. When all recovery times are significantly large, the recoverable-CTP becomes the classic CTP.

The k-CTP is another variant of the CTP, where k is a parameter that represents the maximum number of potential road blockages. When k equals the number of edges, the k-CTP becomes the classic CTP [11]. In the CTP, the traveller selects a path and starts to travel without knowing future blockages, and when he/she encounters a blocked edge must determine whether to wait for the blocked edge to reopen or to look for another way. The main factor to consider is the recovery time versus the time to travel along another path. In this respect, if the problem structure becomes offline instead of online, Bar-Noy and Schieber state that, the optimal strategy is given by the shortest path from source to destination [11]. In the literature, no mathematical model has been developed for the CTP.

### 2.3. Emergency relief transportation literature

Ozdamar and Ertem provide a survey on the response and recovery phases of the disaster management [13]. In disaster management literature, activities of disaster operations management, such as emergency rescue and medical care are categorized in the phase of response, whereas the debris cleanup activities are categorized in the recovery phase [14]. There are a few articles which consider the debris removal as a prerequisite to transport emergency rescue in the response phase.

A study conducted by Berkoune et al. points out a transportation problem which focuses on the emergency aid supply in disaster response phase. They do not consider any road blockages due to debris. In their problem definition, there are a predetermined number of distribution centers, delivery points and groups of humanitarian goods. Their mathematical formulation is not able to provide exact solutions for large scaled instances; therefore, they also introduce genetic algorithm based heuristics [15].

Campbell argues that the structure of emergency response is different from those classical routing problems, as it is not suitable to apply classical cost-minimizing routing aspects. Therefore, the author provides two alternative objective functions one of which minimizes the maximum arrival time, and the other one minimizes the average arrival time. TSP and VRP's with aforementioned objectives are examined, with some extensions on the number of vehicles and vehicle capacity [16].

Another study that emphasizes the distinctness of emergency response structure from classical routing problems, is conducted by Luis et al., where they review the vehicle routing problems on emergency aid transportation in the disaster affected region. They also review some articles where there is uncertainty in demand and supply. Several of these types of problems are solved by using the two-stage stochastic programming models. Classifications due to modeling of vehicle depots, commodities and delivery locations, vehicle fleet types and uncertainty in routes and vehicle fleets also exist in this study [17].

A study conducted by Nolz et al. focus on delivering disaster relief supplies in a disaster affected region, by considering risk which is due to damages on the infrastructure that may prohibit the delivery operation. They formulate the problem as a multi objective optimization program, where the objectives are related with the measures of risk, relief supply coverage and the total travel time. Note that, suggested approaches are using the topographical and geographical properties to measure the risk, and they do not attempt to dispel the causes of risks or the blockages [18].

The most relevant study to our work considering blockages on the roads is conducted by Ozdamar et al. with a focus in providing coordination of debris cleanup operations in a post disaster environment. In their problem structure every road has a blockage, albeit at different intensity, and a blockage is cleaned up by one vehicle at a time. Both deterministic and probabilistic versions of cleanup times are examined. Two goals are considered, one imaximizing the cumulative network accessibility during the debris clearing, and the other one minimizing time to debris clearing. They formulate this NP-hard problem as a recursive Mixed Integer Program and suggest some heuristic methodologies [19]. Note that the study focuses on debris removal rather than emergency relief transportation.

### 2.4. The debris removal problem in the response phase and its relationship to the literature

The roads blocked by debris in the DRP can be considered as the required edge set in the arc-routing literature. However, in the DRP it is not an obligation to unblock all blocked edges if the demand by the critical nodes can be satisfied without doing so. This aspect differentiates the DRP from general routing problems. Another differentiating characteristic of the DRP is the cost definition for the edges: the travel times are defined for all edges in the graph, whereas the effort values are defined only for blocked edges. For the blocked edges, both travel time and service time occur and the edges that are once unblocked incur only the travel time.

Within the node-routing literature, the recoverable-CTP is the problem most resembling the DRP; however, there are significant differences. In the CTP, there is no special effort necessary to unblock roads. A traveller may wait for a road to become open again without doing anything or he/she may find an alternative way. In the DRP, road-unblocking operations are very important. Unlike the recovery times in the CTP, the effort to unblock an edge in the DRP is a function of the amount of debris on it and is independent from the node it is adjacent to. Once an arc is opened, it remains open, and after a blockage on an arc has been resolved, it is possible to benefit from the advantage of re-using this arc, which is the most important distinguishing characteristic of the DRP. Moreover, in the DRP there is one source and many destinations, unlike the CTP's one-destination structure.

In disaster management literature, there are studies which consider emergency relief transportation and debris removal operations, however, none of them deal with dispelling debris blockages in order to transport relief materials. A typical focus in the literature is on post-disaster characteristics of the environment and the adaptation of the well-known routing problems from the literature. Furthermore, the studies considering road blockages do not deal with unblocking operations or transporting emergency relief materials to the critical regions.

In conclusion, even though the DRP shares similar characteristics with arc-routing and node routing problems in the literature, it also has distinguishing characteristics to enable its introduction to the literature as a new variant of the general routing problem.

## 3. Model development

Consider a disaster-affected region as an undirected, complete graph. Districts compose the nodes and roads compose the edges. Districts that require assistance are the critical nodes and the district that involves a qualified supply unit is the supplier. The aim is to provide assistance to critical nodes as soon as possible by travelling along a path that may include blocked edges to be unblocked. To do so, the vehicle RESCUE departs from the supply node and accesses the critical nodes by removing debris on its critical path. Let $G=(N,E)$ be a network where $N$ represents the nodes and $E$ represents the edges. $A=\{(i,j)\cup(j,i):\{i,j\}\in E\}$ constitutes the arc set of the network. It is worth noting that even if the arcs are directed, the parameter settings of arcs $(i,j)$ and $(j,i)$ are symmetric. The node set contains the supply node, critical nodes and potential intermediate nodes. Some edges in the edge set are assumed to be blocked, and are represented by the binary parameter $I_{kl}$, which takes the value of 0 if the arc $(k,l)$ is blocked by debris, and takes the value of 1 otherwise. $t_{kl}$ is the required time to traverse through arc $(k,l)\in A$ and $W_{kl}$ is the required effort in terms of time to remove the debris on edge $\{k,l\}\in E$ if this arc is blocked. Since the parameter settings for arcs $(i,j)$ and $(j,i)$ are symmetric, if the debris on the edge $\{i,j\}\in E$ is removed, arcs $(i,j)$ and $(j,i)$ become available in $A$. Let $DL\subseteq N$ be the set of critical nodes and $SL\subseteq N\backslash DL$ be the set of the chosen supply node and let $DL\cup SL=L$.

### 3.1. Mathematical model

The proposed model determines the visiting order of critical nodes and the travel path between two consecutive critical nodes by also considering blocked roads. The objective of the model is to minimize the total effort spent on travelling along paths and on removing debris on the blocked edges of these paths. The model separately considers travel and debris removal efforts. We define the following decision variables: $TT$: Total travel time until all critical nodes have been visited. $Y_{ij}=1$ if vehicle visits the critical node $j\in L$ right after the critical node $i\in L$, and 0 otherwise. $X_{ijkl}=1$ if the vehicle uses arc $(k,l)\in A$ while traversing from critical node $i\in L$ to critical node $j\in L$, and 0 otherwise. $C_{ij}$ is the cost (time) of traveling from critical node $i\in L$ to critical node $j\in L$, solely in terms of the traversal time. That is, the time effort to remove debris, if necessary, is not included in this value. $B_{kl}=1$ if the debris on edge $\{k,l\}\in E$ is removed, and 0 otherwise. Finally, $P_i$ stands for the visiting time of critical node $i\in DL$ (again excluding the debris removal time). Table 1 summarizes necessary notation.

The mathematical model that minimizes the total effort used until all critical nodes have been visited is as follows:

$$\text{minimize} \quad TT+\sum_{\{k,l\}\in E} B_{kl}W_{kl} \tag{1}$$

$$\text{subject to} \quad \sum_{j\in DL\cup SL} Y_{ji}=1 \quad \forall i\in DL \tag{2}$$

$$\sum_{j\in DL\cup SL} Y_{ij}=1 \quad \forall i\in DL \tag{3}$$

$$\sum_{j\in DL} Y_{ij}=1 \quad \forall i\in SL \tag{4}$$

$$\sum_{l\in N} X_{ijil}-\sum_{l\in N} X_{ijli}=Y_{ij} \quad \forall i,j\in L \tag{5}$$

$$\sum_{l\in N} X_{ijjl}-\sum_{l\in N} X_{ijlj}=-Y_{ij} \quad \forall i,j\in L \tag{6}$$

$$\sum_{l\in N} X_{ijkl}-\sum_{l\in N} X_{ijlk}=0 \quad \forall i,j\in L, k\in N, k\neq i, k\neq j \tag{7}$$

$$P_i=0 \quad \forall i\in SL \tag{8}$$

$$P_j\geq P_i+C_{ij}-M(1-Y_{ij}) \quad \forall i\in L, j\in DL \tag{9}$$

$$TT\geq P_i \quad \forall i\in DL \tag{10}$$

$$\sum_{k,l\in N} X_{ijkl}\leq Y_{ij}\,|N|\,|N| \quad \forall i,j\in L \tag{11}$$

$$C_{ij}=\sum_{k,l\in N} X_{ijkl}t_{kl} \quad \forall i,j\in L \tag{12}$$

**Table 1**
Notation table.

| | |
|---|---|
| Parameters | |
| $I_{kl}$ | 0, if arc $(k,l)$ is blocked by debris;<br>1, otherwise |
| $t_{kl}$ | required time to traverse through arc $(k,l)\in A$ |
| $W_{kl}$ | required effort in terms of time to remove debris<br>on edge $\{k,l\}\in E$ if this arc is blocked |
| Decision Variables | |
| $TT$ | Total travel time until all critical nodes have been visited |
| $Y_{ij}$ | 1, if vehicle visits the critical node $j\in L$ right<br>after the critical node $i\in L$;<br>0 otherwise |
| $X_{ijkl}$ | 1, if vehicle uses arc $(k,l)\in A$ while traversing<br>from the critical node $i\in L$ to critical node $j\in L$;<br>0 otherwise |
| $C_{ij}$ | cost (time) of travelling from critical node $i\in L$<br>to critical node $j\in L$, solely in terms of traversal time |
| $B_{kl}$ | 1, if the debris on edge $\{k,l\}\in E$ is removed;<br>0 otherwise |
| $P_i$ | visiting time of critical node $i\in DL$<br>(excluding the debris removal time) |

$$\sum_{i,j\in DL\cup SL} X_{ijkl}+\sum_{i,j\in DL\cup SL} X_{ijlk}\leq (B_{kl}+I_{kl})\,|L|\,|L| \quad \forall\{k,l\}\in E \tag{13}$$

$$TT\geq 0 \tag{14}$$

$$P_i\geq 0 \quad \forall i\in L \tag{15}$$

$$C_{ij}\geq 0 \quad \forall i,j\in L \tag{16}$$

$$X_{ijkl}\in\{0,1\} \quad \forall i,j\in L,(k,l)\in A \tag{17}$$

$$B_{kl}\in\{0,1\} \quad \forall\{k,l\}\in E \tag{18}$$

$$Y_{ij}\in\{0,1\} \quad \forall i,j\in L \tag{19}$$

Objective (1) minimizes the total travelling time plus the total time spent on debris removal operations until all critical nodes have been visited.

Constraints (2) and (3) together form a visiting order for the critical nodes, which starts and ends at the supply node and ensures visiting each critical node, one after another. It is worth noting that even if the constraints imply that the vehicle returns to the supply node, the objective function of the problem considers the path until all critical nodes have been visited. Constraint (4) ensures that the vehicle visits exactly one critical node right after it departs from the supply node. Constraints (5), (6) and (7) establish a directed path between two consecutive critical nodes, where the directed path is free to include intermediate non-critical nodes. Constraint (8) implies that the vehicle is positioned on the supply node at the beginning. Constraint (9) assigns the visiting time of critical nodes without considering the time spent to remove debris on the blocked edges, if any. Debris removal efforts are taken into account by the objective function. Additionally, Constraint (9) eliminates subtours between critical nodes, and it is worth noting that subtours are allowed between intermediate nodes appearing on different critical path segments. The objective function together with Constraint (10) minimize the most disadvantageous node's visiting time. Constraint (11) guarantees that if there is no visit between a pair of critical nodes, there is no directed path between them. Constraint (12) correctly calculates the total time spent to travel. Constraint (13) guarantees that it is possible to travel along an arc if it is already open or if the debris on it has been removed. Constraints (14)–(19) are the domain restrictions.

The proposed mathematical model has $O(n^4)$ variables and $O(n^3)$ constraints, where $n=|N|$ is the number of nodes.

Even if DRP is a variant of the GRP, as it is mentioned in Section 2.3, the cost definitions for the edges of DRP differ from the classical GRPs, and there is a special effort to unblocking an edge. The above mathematical program developed for DRP seeks for a vehicle route minimizing the total travelling time and the total time spent for debris removing operations. Though the critical nodes in DRP can be viewed as the required nodes in GRP, there are no required edges in DRP as there are in GRP. As it is emphasized before, it is not an obligation to unblock all the blocked edges, if the required nodes can be visited without doing so. Additionally, one of the important characteristics of DRP is the reusing advantage of a blocked edge, once it is opened. In order to handle all these features, unlike the models present in the litrature, our model needs to keep track of the tour as a union of path segments between critical nodes allowing for different path segments share the same edge and necessitating the use of 4-index decision variables.

## 4. Heuristic solution methodology

Our experimentation with the mathematical model has shown that as the network dimensions and the number of critical nodes increase, it becomes harder to reach the optimal solution in a reasonable amount of time. It may even take hours to find the optimum solution for certain instances. However, by the nature of the problem an immediate decision might be necessary. Therefore, we decided to develop a heuristic methodology within the scope of solution quality versus time trade-off, which can find good solutions expeditiously without straying too far from the optimal solution.

For that purpose, we developed a fast constructive heuristic solution methodology called the *initial route heuristic*, based on Dijkstra's shortest-path algorithm. For better optimality gaps, we also applied an improvement heuristic methodology, which can be considered as a variation of the 2-opt algorithm [20].

Our initial route heuristic starts from the source node and applies Dijkstra's algorithm until all critical nodes have been visited. In this context, the algorithm first finds the shortest-path tree that is rooted at the source node to other nodes until a critical node, say $j$, is reached. Then, having found the closest critical node $j$, the algorithm travels along the path from the source to node $j$. It unblocks the blocked edges on this path, if any. Since after a blocked edge is opened it remains open, the debris removal cost for such an edge is not paid if it is used again. Then, the algorithm considers node $j$ as the new source node and applies the same algorithm again and again until all critical nodes have been visited. The objective of the algorithm is to find a path for RESCUE with minimum effort until all critical nodes have been visited. This effort involves both travel time and edge unblocking.

The flow chart of the algorithm is depicted in Appendix A.

To improve the solution quality, by having the output of the initial route heuristic as the input, we applied the 2-opt algorithm. Let $start \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_{h-1} \rightarrow i_h \rightarrow \dots \rightarrow i_{e-1} \rightarrow i_e \rightarrow \dots \rightarrow i_k$ be the output path of the constructive heuristic. Note that this path includes all critical nodes as well as some non-critical ones. The 2-opt algorithm randomly selects two nodes, say, $i_h$ and $i_e$, from this path. It preserves the same order for the nodes from start to $i_{h-1}$ and from $i_{e+1}$ to the end. It reverses the order for the nodes from $i_h$ to $i_e$. An initial route which is used as an input is shown in Fig. 1 and the resulting order after 2-opt algorithm applied is shown in Fig. 2, for the data of Kartal. By applying this procedure until no improvement is obtained, we get another path which contains the same nodes but in a different order with a better objective value. The objective function, which is the total cost of the route is recalculated for each path since the 2-opt algorithm may possibly replace a blocked arc with an unblocked arc or vice versa. If a blocked arc is included in the resulting route, the debris is removed to unblock this arc and corresponding debris removal cost is added to the objective. It is worth noting that, the re-blockage of arcs is not possible. Therefore, after the blockage of an arc is resolved, it remains open forever. To represent this issue in the model, the $I$ matrix is updated in the way that the 0 value of the arc, whose blockage is removed, is changed into 1.

The flow chart is given in Appendix B.

These types of heuristics are applicable for different kinds of routing problems, as well. For example, the heuristic may be adapted to hazmat transportation problems by constructing an analogy between road blockages and riskiness of the arcs.

In terms of theoretical bounds of heuristics, there are some remarks about 2-opt algorithm for TSP in the literature. It is indicated that, given a bad initial starting tour, a ratio of at least $1/4N$ is guaranteed as the best performance when triangle inequality holds. Also it is emphasized that, it does not yield a ratio worse than $4N$. Additionally, it is given that, if the initial starting tour is generated by some smart choice, the worst case behaviors show significant improvement [21].

In this context, even DRP and TSP are different problems, it can be said that, the theoretical bounds of 2-opt algorithm for our problem possibly reflect similar results with the ones mentioned above, taking account of triangle inequality and the initial starting tour given by the constructive heuristic.

## 5. Data and computational results

### 5.1. Data

To measure the effectiveness of the developed model and the heuristic methodology we used a data set from Turkey based on Istanbul's Kartal municipality [22]. We examined the districts and selected those that contain schools or hospitals as critical ones, which amount to seven. The Marmara Region Disaster Center of the Turkish Red Crescent, which is located in Kartal, serves as the supplier.

Table 2 summarizes the features of this data set.

The map in Fig. 3 shows the locations of the supplier and the critical nodes in Kartal, where the red triangle represents the supplier, yellow circles illustrate the critical nodes with schools and green circles illustrate the critical nodes with hospitals. Red circles represent other nodes which are neither supplier nor critical.

We used the node-to-node distance matrix of the Kartal data set to calculate the travel time of RESCUE between nodes $k$ and $l$, namely, a $t_{kl}$ matrix. We assumed that vehicle speed is about 20 km/h, and by dividing the distances by this speed, formed the $t_{kl}$ matrix. We applied the Floyd-Warshall algorithm [23] so that the $T$=[$t_{kl}$] matrix is symmetric and satisfies triangle inequality. For constituting the $I_{kl}$ matrix, which indicates the arc blockages, and the $W_{kl}$ matrix, which provides the required times to remove arc debris, we categorized potential earthquakes according to severity. To this end, we constituted four groups of severities. We varied the severity of an earthquake (SOE) from 1 to 4, where (4) is the most severe. Table 3 illustrates the intervals of edge-blockage ratios according to the SOE. Using this classification and the corresponding blocked-edge ratio (BER) values given in Table 3, blockages were randomly assigned to the arcs of the $I_{kl}$ matrix. Obviously, there are more blocked edges for more-severe earthquakes.

Since the required effort to remove debris from blocked edges is directly related to the SOE, the required effort is calculated so as to have higher values for more-severe earthquakes. To observe how the debris removal effort affects the computational results, we calculate $W_{kl}$ in two different ways. In both methods the values are proportional to the SOE and the length of the relevant edge, but one uses more debris removal effort. The following equations show how $W_{kl}$ is calculated:

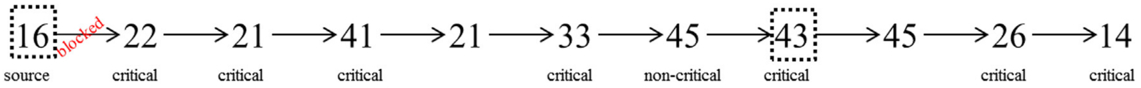


**Fig. 1.** Input of 2-opt algorithm.

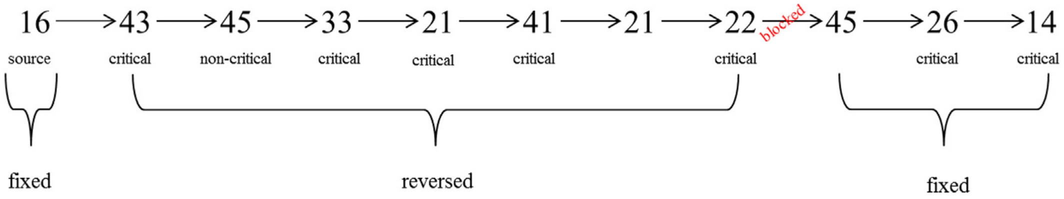


**Fig. 2.** Illustrative example of 2-opt algorithm.

$$W_{kl} = SOE{*}t_{kl} + U\left[0, \max_{ij} t_{ij}\right]$$

$$W'_{kl} = SOE{*}t_{kl}$$

In this context, we created 20 different instances with the Kartal data set. For each class of SOE there exists five different instances whose blocked-edge ratios - and accordingly the number of blocked edges - are equivalent, but the edges that are blocked are different. Additionally, to see the effect of the debris removal effort on the results, these 20 instances are considered with two different debris removal effort ($W_{kl}$, $W'_{kl}$ ) values, which are calculated as mentioned above. We also vary the number of critical nodes by selecting only hospitals, only schools and both hospitals and schools. Table 4 depicts the SOE, BER and number of blocked arc settings of the instances used in the computational experiments. It is worth noting that when SOE = 1, the number of blocked arcs for all computational experiments in the Kartal data set is 124, however, their locations differ. The same is true for other classes of SOE. To test the effects of this issue, we selected five instances for a class of SOE. Namely, when SOE = 1, the corresponding instances are $K-1$,$K-2$...$K-5$. In each row, the instances where the corresponding debris removal effort is greater, ($W_{kl}$) is shown on the top, such as $K-1$...$K-5$; whereas the instances where the debris removal effort is smaller, ($W'_{kl}$) is shown below, such as ($K-1'$…$K-5$').

### 5.2. Computational analysis

In this subsection, we discuss the computational results of the mathematical model and the heuristic. The computational experiments of the mathematical model were conducted with CPLEX 12.4 on a 4 x AMD Opteron Interlagos 16C 6282SE 2.6G 16M 6400MT computer, running under the Linux operating system. The heuristic algorithms were coded in Java 1.6.0-23 on the same computer.

The computational results are summarized in the Tables 5–8. Tables 5 and 6 illustrate the model performances. The tables show the total effort spent to visit all the critical nodes in the optimum solution for the instances that the optimum is achieved within the 4-h time limit, the CPU times and the arcs where debris is removed in the resulting solution. The names of the instances are stated in the second column of the tables and the corresponding settings given in the Table 4 are used for each instance group. Table 5 shows the Kartal instances with $W_{ij}$, and the results of the instances with $W'_{ij}$ are summarized in Table 6. The remaining tables are organized to analyze the heuristic performances.

**Table 2**
Features of the data set. (Numbers in bold indicate node numbers.)

| | Kartal |
|---|---|
| No. of nodes | 45 |
| Symmetric distance matrix and triangle inequality requirement | Yes |
| Supply node (**node number**, name of place) | **16**-Marmara region disaster center of Turkish Red Crescent |
| Total no. of critical nodes | 7 |
| No. of schools | 3 |
| (**node numbers**) | **14,21,22** |
| No. of hospitals | 4 |
| (**node numbers**) | **26,33,41,43** |

Table 5 summarizes the computational results for the Kartal instances where the debris removal effort is greater. The table is divided into four parts, where each part illustrates the results of the five sets of instances belonging to a class of SOE, as reported in Table 4. In each part, for a class of SOE; the results for different amounts of critical nodes are illustrated. For example, for the instances $K-1$,…,$K-5$, the earthquake severity is 1, that is, SOE = 1, and experiments are conducted for seven (all), four (hospitals) and three (schools) critical nodes for this SOE class. Additionally, since the instances $K-1$,…,$K-5$ differ in terms of the locations of the blocked edges, we repeated each experiment five times for a fixed critical node set. To clarify, when the number of critical nodes is seven, the corresponding five rows in the table coincide with the results of the five instances, $K-1$,…,$K-5$, respectively. Concisely, Tables 5 and 6 illustrate the computational results of the Kartal data set, where one of the two debris removal effort settings is depicted in each table.

Since the number of blocked edges increases as the SOE grows, it becomes compulsory to unblock edges to visit critical nodes for some instances with greater SOE values. When the experiments are repeated for identical instances with different debris removal effort requirements, such as $K-11$ and $K-11'$, we observe that RESCUE does not hesitate to travel on a path that includes blocked arcs when the required effort to unblock them is relatively low. As indicated above, for a class of SOE, we repeat our experiment with five different instances, where the blocked-edge ratios and the number of blocked edges are the same, but the edges that are blocked are different.

Regarding CPU times, this issue has a marginal effect when the number of critical nodes remains constant. Namely, for the instances $K-1$,$K-2$,$K-3$,$K-4$ and $K-5$, when the number of critical nodes is seven, CPU times range from 183 s to 221 s; for four critical nodes, the range is between six and 12 s, and when it is required to visit only three critical nodes, all instances are solved to optimality within 1 s.

However, as the number of critical nodes increases, CPU times exponentially increase. For instance, in $K-1$, the CPU time is 1.4 s for three critical nodes, 8.81 s for four critical nodes and 221.41 s when the critical node number scales up to seven. Additionally, together with the number of critical nodes to visit, the SOE significantly affects CPU time. As expected, for more-severe earthquakes, solution times increase. To exemplify, when the SOE = 3, the instance $K-11'$ is solved in 214.86 s for seven critical nodes, but as the SOE increases to four, the solution time of the instance $K-16'$ for seven critical nodes, scales up to 4864.74 s. Both $K-11'$, and $K-16'$ intend to visit seven critical nodes, but due to the differences in the corresponding SOEs, the numbers of blocked edges differ.

Also, when we analyze the CPU times for the instances where only the debris removal effort is changed, such as $K-16$ and $K-16'$, we observe that the CPU times for instances with smaller debris removal efforts are smaller. This issue is apparent for the instances

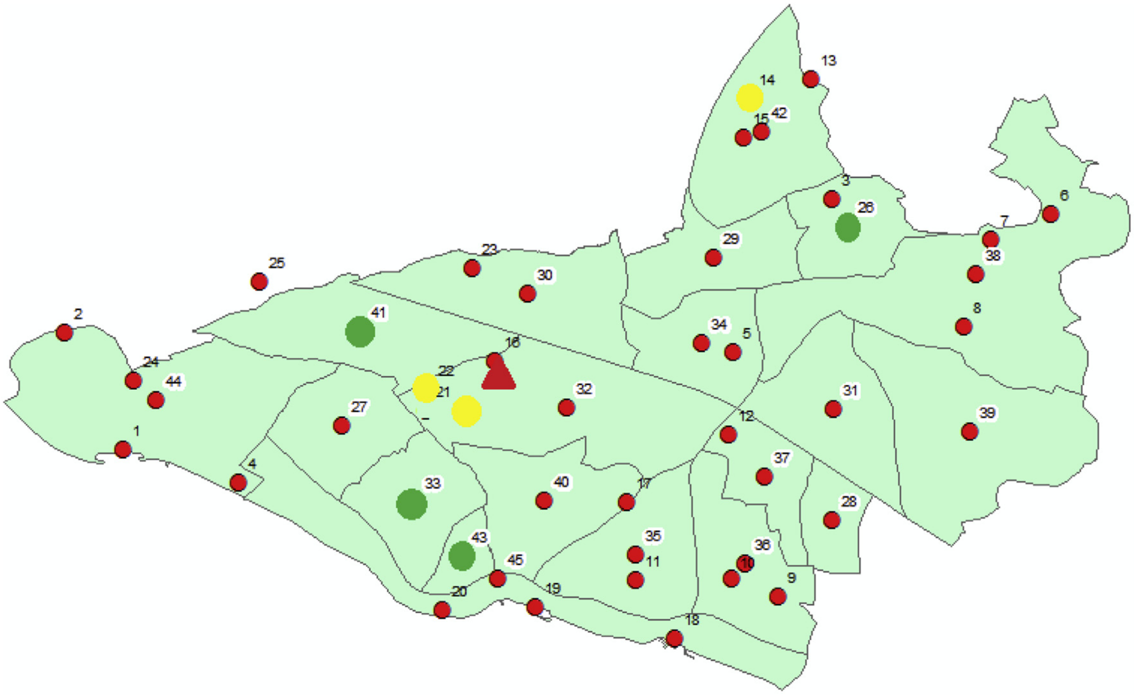


**Fig. 3.** The location of supplier and critical nodes in Kartal.

**Table 3**
SOE and corresponding BER values.

| Severity of earthquake (SOE) | Blocked edge ratio (BER) |
|---|---|
| SOE = 1 | 0%–20% |
| SOE = 2 | 20%–50% |
| SOE = 3 | 50%–80% |
| SOE = 4 | 80%–100% |

where the chance of encountering a blocked edge increases, namely, as the SOE and the number of critical nodes increase.

Additionally, when the objective values of the instances from Tables 5 and 6 are compared, we see that the objective values are the same for the instances from both tables when there is no need to remove debris from an edge; however, the objective values of the instances with smaller debris removal efforts are smaller when the debris from an edge is removed. For example, for $K\!-\!1$ and $K\!-\!1'$, the objective values are similar for all numbers of critical nodes because there are no debris-removed edges for these instances. However, for the instance $K\!-\!7'$, with seven critical nodes, the objective is 49 and the arc (33, 43) is unblocked. When we investigate the same instance with greater debris removal effort, $K\!-\!7$, we see that the objective is 50, and for no arc the debris removal operation is completed.

**Table 4**
Kartal instances and the corresponding SOE, BER and number of blocked-edge settings.

| Removal effort | Kartal instances | SOE | BER | No. of blocked arcs |
|---|---|---|---|---|
| $W_{kl}$ | $K$−1...$K$−5 | 1 | 0.125 | 124 |
| $W'_{kl}$ | $K$−1'...$K$−5' | | | |
| $W_{kl}$ | $K$−6...$K$−10 | 2 | 0.445 | 441 |
| $W'_{kl}$ | $K$−6'...$K$−10' | | | |
| $W_{kl}$ | $K$−11...$K$−15 | 3 | 0.58 | 574 |
| $W'_{kl}$ | $K$−11'...$K$−15' | | | |
| $W_{kl}$ | $K$−16...$K$−20 | 4 | 0.819 | 806 |
| $W'_{kl}$ | $K$−16'...$K$−20' | | | |

When we analyze the resulting travel paths for instances with the same number of blocked edges for different locations, we realize that generally, the locations of critical nodes are more important than the locations of the blocked edges for less-severe earthquakes. For example, we observe from the resulting travel paths from the solutions of instances $K\!-\!1$, $K\!-\!2$, $K\!-\!3$, $K\!-\!4$ and $K\!-\!5$ that even though the vehicle follows different travel paths for these instances due to the different locations of blocked edges, the order of visiting the critical nodes remains the same. However, as the SOE increases, the number of blocked edges also increases, and the order of visiting the critical nodes differs greatly, as in the case with $K\!-\!16$,$K\!-\!17$,$K\!-\!18$,$K\!-\!19$ and $K\!-\!20$.

As the network dimenstions increase, it becomes difficult to find the optimum solutions for some instances. For this reason we developed our heuristic methodology that quickly reaches good, feasible solutions for this problem. First, we implement an initial route algorithm, then apply a 2-opt improvement phase. The method's results are summarized in Tables 7 and 8 The solution times of the heuristic are less than a second for each instance, therefore we do not report them.

Since, the mathematical model can find optimal solutions for all Kartal instances, it is possible to compare the heuristic performances with the optimum solutions of the Kartal instances. Table 7 summarizes the heuristics performances where the results are obtained from analyzing all classes of SOE, namely, all instances from $K\!-\!1$ to $K\!-\!20$ and $K\!-\!1'$ to $K\!-\!20'$, with different numbers of critical nodes. The first column under Overall Heuristic Results indicates the average gap of the heuristic solution from the optimum solution for the five instances. The second columns stands for the maximum gap value over all instances, and the optimum ratio of the number of instances solved to optimality with the heuristic is illustrated in the third column. Specifically, the first row of the table indicates that the heuristic finds the optimum for 80% of the Kartal instances with $W_{ij}$, where $|DL|$ = seven. Since there are 20 such instances, our heuristic finds the optimum for 16 of them. As is evident from the table, our heuristic methodology can find

**Table 5**
Model performances of Kartal instances with $W_{ij}$.

| No.of critical nodes | Instances | Best objective | Cplex CPU(sec) | Debris removed arcs | Instances | Best objective | Cplex CPU(sec) | Debris removed arcs |
|---|---|---|---|---|---|---|---|---|
| | SOE = 1 | | | | SOE = 2 | | | |
| 7 | K-1 | 44 | 221.41 | – | K-6 | 48 | 223.26 | – |
| (all) | K-2 | 43 | 186.73 | – | K-7 | 50 | 235.96 | – |
| | K-3 | 44 | 190.82 | – | K-8 | 51 | 270.41 | – |
| | K-4 | 43 | 178.4 | – | K-9 | 49 | 225.46 | – |
| | K-5 | 43 | 183.2 | – | K-10 | 48 | 261.39 | – |
| 4 | K-1 | 35 | 8.81 | – | K-6 | 38 | 15.39 | – |
| (hospitals) | K-2 | 35 | 10.02 | – | K-7 | 42 | 8.4 | – |
| | K-3 | 36 | 6.27 | – | K-8 | 40 | 9.09 | – |
| | K-4 | 35 | 12.08 | – | K-9 | 42 | 10.75 | – |
| | K-5 | 35 | 11.69 | – | K-10 | 39 | 7.04 | – |
| 3 | K-1 | 30 | 1.4 | – | K-6 | 29 | 2.52 | – |
| (schools) | K-2 | 30 | 1.47 | – | K-7 | 29 | 2.54 | – |
| | K-3 | 30 | 1.61 | – | K-8 | 30 | 2.28 | – |
| | K-4 | 29 | 1.44 | – | K-9 | 35 | 3.51 | – |
| | K-5 | 29 | 1.44 | – | K-10 | 30 | 3.04 | – |
| | SOE = 3 | | | | SOE = 4 | | | |
| 7 | K-11 | 53 | 315.37 | – | K-16 | 109 | 5167.18 | (21,22) |
| (all) | K-12 | 63 | 495.63 | – | K-17 | 82 | 3319.8 | – |
| | K-13 | 68 | 381.4 | – | K-18 | 110 | 9136.87 | (21,22) |
| | K-14 | 46 | 196.96 | – | K-19 | 90 | 2915.86 | (21,22) |
| | K-15 | 47 | 186.72 | – | K-20 | 101 | 4541.84 | (21,22),(22,41),(33,43) |
| 4 | K-11 | 35 | 4.87 | – | K-16 | 84 | 22.32 | – |
| (hospitals) | K-12 | 53 | 9.51 | – | K-17 | 67 | 20.23 | – |
| | K-13 | 51 | 8.32 | – | K-18 | 70 | 38.91 | (33,43) |
| | K-14 | 38 | 8.69 | – | K-19 | 70 | 49.49 | – |
| | K-15 | 40 | 8.41 | – | K-20 | 88 | 64.54 | – |
| 3 | K-11 | 40 | 4.34 | – | K-16 | 57 | 6.62 | (21,22) |
| (schools) | K-12 | 35 | 2.98 | – | K-17 | 55 | 4.66 | – |
| | K-13 | 30 | 3.11 | – | K-18 | 68 | 7.08 | (21,22) |
| | K-14 | 35 | 2.85 | – | K-19 | 55 | 4.76 | (21,22) |
| | K-15 | 29 | 2.51 | – | K-20 | 45 | 3.62 | (21,22) |

**Table 6**
Model performances of Kartal instances with $W'_{ij}$.

| No.of critical nodes | Instances | Best objective | Cplex CPU(sec) | Debris removed arcs | Instances | Best objective | Cplex CPU(sec) | Debris removed arcs |
|---|---|---|---|---|---|---|---|---|
| | SOE = 1 | | | | SOE = 2 | | | |
| 7 | K-1′ | 44 | 219.89 | – | K-6′ | 48 | 231.4 | – |
| (all) | K-2′ | 43 | 213.25 | – | K-7′ | 49 | 210.34 | (33,43) |
| | K-3′ | 44 | 192.61 | – | K-8′ | 51 | 282.31 | – |
| | K-4′ | 43 | 156.37 | – | K-9′ | 49 | 259.62 | – |
| | K-5′ | 43 | 152.9 | – | K-10′ | 48 | 210.25 | – |
| 4 | K-1′ | 35 | 7.16 | – | K-6′ | 38 | 9.2 | – |
| (hospitals) | K-2′ | 35 | 10.12 | – | K-7′ | 41 | 12.63 | (33,34) |
| | K-3′ | 36 | 9.98 | – | K-8′ | 40 | 10.28 | – |
| | K-4′ | 35 | 7.02 | – | K-9′ | 42 | 13.18 | – |
| | K-5′ | 35 | 7.31 | – | K-10′ | 39 | 20.43 | – |
| 3 | K-1′ | 30 | 2.37 | – | K-6′ | 29 | 1.57 | – |
| (schools) | K-2′ | 30 | 1.77 | – | K-7′ | 29 | 2.75 | – |
| | K-3′ | 30 | 2.76 | – | K-8′ | 30 | 3.38 | – |
| | K-4′ | 29 | 1.86 | – | K-9′ | 32 | 3.8 | (21,22) |
| | K-5′ | 29 | 1.85 | – | K-10′ | 30 | 1.9 | – |
| | SOE = 3 | | | | SOE = 4 | | | |
| 7 | K-11′ | 51 | 214.86 | (21,22) | K-16′ | 97 | 4864.74 | (3,26),(21,22),(43,45) |
| (all) | K-12′ | 63 | 316.91 | – | K-17′ | 78 | 2422.78 | (22,41) |
| | K-13′ | 67 | 336.34 | (27,33) | K-18′ | 95 | 3791.44 | (14,15),(21,22),(33,43) |
| | K-14′ | 46 | 198.85 | – | K-19′ | 81 | 2794.34 | (21,22),(43,45) |
| | K-15′ | 47 | 184.67 | – | K-20′ | 80 | 3258.78 | (21,22),(22,41),(33,43) |
| 4 | K-11′ | 35 | 7.84 | – | K-16′ | 80 | 54.89 | (16,32),(43,45) |
| (hospitals) | K-12′ | 53 | 8.37 | – | K-17′ | 67 | 17.18 | (22,41) |
| | K-13′ | 50 | 16.5 | (33,43) | K-18′ | 63 | 32.48 | (33,43) |
| | K-14′ | 38 | 9.42 | – | K-19′ | 70 | 26.91 | – |
| | K-15′ | 40 | 13.71 | – | K-20′ | 73 | 14.96 | (21,22),(22,41),(33,43) |
| 3 | K-11′ | 38 | 3.42 | (21,22) | K-16′ | 50 | 4.88 | (21,22) |
| (schools) | K-12′ | 33 | 3.12 | (21,22) | K-17′ | 51 | 3.48 | (21,22) |
| | K-13′ | 30 | 2.93 | – | K-18′ | 61 | 5.24 | (21,22) |
| | K-14′ | 35 | 2.03 | – | K-19′ | 48 | 4.73 | (21,22) |
| | K-15′ | 29 | 2.65 | – | K-20′ | 38 | 4.51 | (21,22) |

**Table 7**
Heuristic performance summary of Kartal instances.

| Instances:<br>K-1, …,K-20<br>K-1′, …,K-20′ | | Overall heuristic results | | |
|---|---|---|---|---|
| | | Average gap | Maximum gap | Optimum ratio |
| Kartal- with $W_{ij}$ | $\lvert DL \rvert = 7$ | 1.3% | 7.5% | 80% |
| | $\lvert DL \rvert = 4$ | 0.33% | 2.90% | 85% |
| | $\lvert DL \rvert = 3$ | 0.3% | 4.40% | 90% |
| Kartal- with $W'_{ij}$ | $\lvert DL \rvert = 7$ | 2.17% | 18.80% | 65% |
| | $\lvert DL \rvert = 4$ | 1.55% | 13.70% | 75% |
| | $\lvert DL \rvert = 3$ | 0.385% | 5.30% | 90% |

optimum solutions for up to 90% of the Kartal instances. To illustrate the effect of SOE individually on the results, each quarter of the Table 8 considers a set of five instances according to a class of SOE. The first quarter of the table shows instances $K-1$, …,$K-5$ and $K-1'$, …,$K-5$', with the same focus as Table 7, where their corresponding SOE is 1. It is worth noting that Table 7 gives the averages of all SOE classes and in Table 8 they are depicted separately. For each SOE, the highlighted rows report the heuristic performances for Kartal with greater ($W_{ij}$) and smaller ($W'_{ij}$) debris removal efforts individually, arranged over the number of critical nodes.

From the aspect of heuristic performances according to debris removal effort settings, the heuristic results are almost the same for the instances in the first two quarters of the table; however, in the third quarter, it can be seen that the heuristic gives better results where the debris removal effort is greater (87% optimum ratio), and it reduces to 73% for the instances with smaller debris removal efforts. Also, in the fourth quarter, for greater debris removal efforts the heuristic finds the optimum at a rate of 60%, which reduces to 40% for the cases with smaller debris removal effort. That is, as the SOE increases, the heuristic shows better performances for the instances with $W_{ij}$ compared with the instances where the debris removal effort is smaller ($W_{ij}'$). With the latter, the trade-off between unblocking a road and finding alternative paths is minor. However, with greater debris removal efforts on the edges, and for more-severe earthquakes (which imply more blocked edges), the trade-off is obvious. Therefore, this may be why the heuristic gives better results for instances with greater debris removal efforts. When the performances are analyzed from the perspective of number of critical nodes, we see in Table 7 that the optimum ratio is highest when the number of critical nodes is lowest, 3, and the ratio decreases for higher numbers of nodes. However, contrary to this inference, apparent difficulty in finding optimum solutions when the number of critical nodes increases is not valid when the instances are dealt with separately, according to SOE class, as in Table 8. Surprisingly, as we observe in the fourth quarter of this table, this method gives better optimum ratios for seven critical nodes than for the cases with four critical nodes. This result may occur due to the differences among the instances, or, since the

**Table 8**
Heuristic performance summary of Kartal instances for each SOE class.

| | | | | | |
|---|---|---|---|---|---|
| SOE = 1 | Instances: | | Overall heuristic results | | |
| | K-1, … K-5 | | Average gap | Maximum gap | Optimum ratio |
| | K-1′, … K-5′ | | | | |
| | Kartal-with $W_{ij}$ | $\lvert DL \rvert$=7 | 0.9% | 4.5% | 80% |
| | | $\lvert DL \rvert$=4 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=3 | 0.0% | 0.0% | 100% |
| | | | 0.3% | 4.5% | 93% |
| | Kartal-with $W_{ij}'$ | $\lvert DL \rvert$=7 | 0.9% | 4.5% | 80% |
| | | $\lvert DL \rvert$=4 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=3 | 0.0% | 0.0% | 100% |
| | | | 0.3% | 4.5% | 93% |
| SOE = 2 | Instances: | | Overall heuristic results | | |
| | K-6, … K-10 | | Average gap | Maximum gap | Optimum ratio |
| | K-6′, … K-10′ | | | | |
| | Kartal-with $W_{ij}$ | $\lvert DL \rvert$=7 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=4 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=3 | 0.0% | 0.0% | 100% |
| | | | 0.0% | 0.0% | 100% |
| | Kartal-with $W_{ij'}$ | $\lvert DL \rvert$=7 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=4 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=3 | 0.0% | 0.0% | 100% |
| | | | 0.0% | 0.0% | 100% |
| SOE = 3 | Instances: | | Overall heuristic results | | |
| | K-11, … K-15 | | Average gap | Maximum gap | Optimum ratio |
| | K-11′, … K-15′ | | | | |
| | Kartal-with $W_{ij}$ | $\lvert DL \rvert$=7 | 1.52% | 7.5% | 80% |
| | | $\lvert DL \rvert$=4 | 0.0% | 0.0% | 100% |
| | | $\lvert DL \rvert$=3 | 0.52% | 2.5% | 80% |
| | | | 0.68% | 7.5% | 87% |
| | Kartal-with $W_{ij}'$ | $\lvert DL \rvert$=7 | 2.24% | 9.8% | 60% |
| | | $\lvert DL \rvert$=4 | 0.4% | 2.0% | 80% |
| | | $\lvert DL \rvert$=3 | 0.52% | 2.6% | 80% |
| | | | 1.05% | 9.8% | 73% |
| SOE = 4 | Instances: | | Overall heuristic results | | |
| | K-16, … K-20 | | Average | Maximum | Optimum |
| | K-16′, … K-20′ | | gap | gap | ratio |
| | Kartal-with $W_{ij}$ | $\lvert DL \rvert$=7 | 2.68% | 7.9% | 60% |
| | | $\lvert DL \rvert$=4 | 1.32% | 2.9% | 40% |
| | | $\lvert DL \rvert$=3 | 0.82% | 4.4% | 80% |
| | | | 1.60% | 7.9% | 60% |
| | Kartal-with $W_{ij}'$ | $\lvert DL \rvert$=7 | 5.6% | 18.8% | 20% |
| | | $\lvert DL \rvert$=4 | 5.84% | 13.7% | 20% |
| | | $\lvert DL \rvert$=3 | 1.06% | 5.3% | 80% |
| | | | 4.16% | 18.8% | 40% |

number of critical nodes (seven, four and three) are not quantitatively very different from each other, the locations of the critical nodes could be the determining factor for the heuristic performances. Finally, from the perspective of how SOE affects the heuristic performances, the optimum finding rate is higher for less-severe earthquakes and lower when SOE = 3 and SOE = 4, as expected.

In conclusion, we observe that the severity of an earthquake and the number of critical nodes are the main factors that affect the performances of both the mathematical model and the heuristic. In other respects, we observe that the locations of the critical nodes affect the optimum travel path, and that, the heuristic performances are influenced by the locations of the critical nodes.

Finally, in order to come up with some managerial insights, Tables 5 and 6 are analyzed. It is worth noting that data is randomly generated, and the blockages are also random. However, the occurrence number of arcs (21,22) and (33,43) on the debris removed arc columns of both tables is remarkable. In Table 5, the only edges where the debris is removed are (21,22), (22,41) and (33,43), which occurs when SOE is four. Nodes 21, 22, 41 and 43 are all critical nodes, where 21 and 22 are schools; 33, 41 and 43 are hospitals. Additionally, when their geographical position is considered, it is observed that, nodes 21 and 22; 22 and 41; and 33 and 43 are very close to each other. Therefore, it is meaningful to remove debris on these arcs, if they are blocked, which is the case here. In Table 6, on the debris removed arcs column, the arcs (33,43), (33,34), (21,22), (3,26), (43,45), (22,41), (14,15), (16,32) and (27,33) are shown. For each of these arcs, at least one of the nodes are either critical or the supplier (node 16 represents the supply node). Namely, arc (16,32) represents a transfer from supplier to a non-critical node, and other arcs either represent a transfer between two critical nodes, or an arrival to a critical node from a non-critical one, or a departure from a critical node to a non-critical node. In terms of their geographical closeness, except the node pair 33 and 34, others are quite close to each other. So, we can say that, it is important to unblock the edges between critical nodes or which are incident to a critical node, especially if they are close to each other.

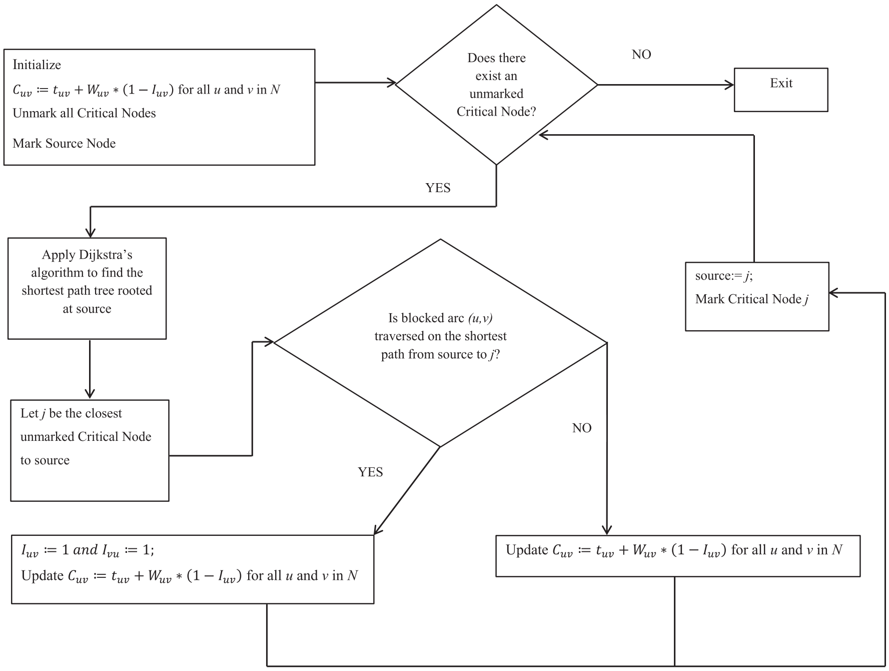
## 6. Conclusion

Due to the importance of emergency aid transportation during the post-earthquake response phase, in this study, we developed a solution methodology that provides emergency supplies to predetermined disaster areas by considering blockages on the transportation network. The main contribution of the proposed methodology is to enable providing disaster-relief materials to areas as soon as possible, thus saving lives and defusing the chaotic post-disaster environment. The problem characteristics imply both a node-routing aspect, requiring the vehicle to visit predetermined disaster areas, and an arc-routing aspect, where it may be necessary to unblock some of the arcs on the vehicle's travel path. For this reason, we studied both the arc-routing and node-routing literature, which are under the umbrella of the general routing literature. Then we mathematically modeled the problem to minimize the total effort spent until all critical nodes were visited, where ”total effort” includes both travelling and debris removal efforts. The model assigns the visiting order of the critical nodes and decides the travelling path between them, indicating the arcs with blockage to be removed. Our heuristic methodology mitigates the solution time difficulties encountered when the dimension of the network increases. The performances of the model and the heuristic were tested with a data set from Istanbul's Kartal district.

## Acknowledgments

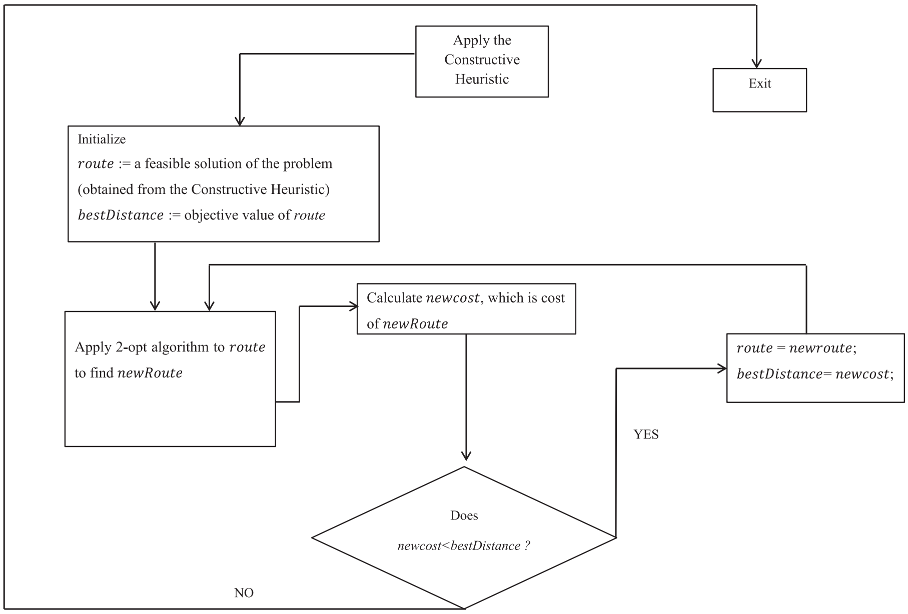
This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under the grant number 112M225. We also acknowledge the contribution of the editor and the reviewers in leading to this improved current form of the article.

## Appendix A


Initialize
$C_{uv} := t_{uv} + W_{uv} * (1 - I_{uv})$ for all $u$ and $v$ in $N$
Unmark all Critical Nodes
Mark Source Node
Does there exist an unmarked Critical Node?
NO
Exit
YES
Apply Dijkstra's algorithm to find the shortest path tree rooted at source
source:= $j$;
Mark Critical Node $j$
Is blocked arc $(u,v)$ traversed on the shortest path from source to $j$?
Let $j$ be the closest unmarked Critical Node to source
NO
YES
$I_{uv} := 1$ and $I_{vu} := 1$;
Update $C_{uv} := t_{uv} + W_{uv} * (1 - I_{uv})$ for all $u$ and $v$ in $N$
Update $C_{uv} := t_{uv} + W_{uv} * (1 - I_{uv})$ for all $u$ and $v$ in $N$

Flow chart of the constructive heuristic.

## Appendix B


Apply the Constructive Heuristic
Exit
Initialize
*route* := a feasible solution of the problem
(obtained from the Constructive Heuristic)
*bestDistance* := objective value of *route*
Apply 2-opt algorithm to *route* to find *newRoute*
Calculate *newcost*, which is cost of *newRoute*
*route* = *newroute*;
*bestDistance*= *newcost*;
YES
Does
*newcost*<*bestDistance* ?
NO

Flow chart of the improvement heuristic with 2-opt.

## References

[1] Karp RM. Reducibility among combinatorial problems. Springer; 1972.

[2] Lenstra JK, Kan A. On general routing problems. Networks 1976;6(3):273–80.

[3] Corberán A, Sanchis J. A polyhedral approach to the rural postman problem. Eur J Operational Res 1994;79(1):95–114.

[4] Kwan M-K. Graphic programming using odd or even points. Chin Math 1962;1(273–277):110.

[5] Wøhlk S. A decade of capacitated arc routing. In: The vehicle routing problem: latest advances and new challenges. Springer; 2008. p. 29–48.

[6] Orloff C. A fundamental problem in vehicle routing. Networks 1974;4(1): 35–64.

[7] Eiselt HA, Gendreau M, Laporte G. Arc routing problems, part i: the chinese postman problem. Operations Res 1995;43(2):231–42.

[8] Golden BL, Wong RT. Capacitated arc routing problems. Networks 1981;11(3): 305–15.

[9] Xu Y, Hu M, Su B, Zhu B, Zhu Z. The canadian traveller problem and its competitive analysis. J Comb Optim 2009;18(2):195–205.

[10] Papadimitriou CH, Yannakakis M. Shortest paths without a map. In: Automata, languages and programming. Springer; 1989. p. 610–20.

[11] Bar-Noy A, Schieber B. The canadian traveller problem, in: proceedings of the second annual ACM-SIAM symposium on discrete algorithms. Soc Industrial Appl Math 1991:261–70.

[12] Nikolova E, Karger DR. Route planning under uncertainty: the canadian traveller problem. In: AAAI; 2008. p. 969–74.

[13] L. Özdamar, M. A. Ertem, Models, solutions and enabling technologies in humanitarian logistics, Eur J Operational Res.

[14] Altay N, Green WG. Or/ms research in disaster operations management. Eur J operational Res 2006;175(1):475–93.

[15] Berkoune D, Renaud J, Rekik M, Ruiz A. Transportation in disaster response operations. Socio-Economic Plan Sci 2012;46(1):23–32.

[16] Campbell AM, Vandenbussche D, Hermann W. Routing for relief efforts. Transp Sci 2008;42(2):127–45.

[17] Luis E, Dolinskaya IS, Smilowitz KR. Disaster relief routing: integrating research and practice. Socio-economic Plan Sci 2012;46(1):88–97.

[18] Nolz PC, Semet F, Doerner KF. Risk approaches for delivering disaster relief supplies. OR Spectr 2011;33(3):543–69.

[19] Özdamar L, Aksu DT, Ergüneş B. Coordinating debris cleanup operations in post disaster road networks. Socio-Economic Plan Sci 2014;48(4):249–62.

[20] Croes G. A method for solving traveling-salesman problems. Operations Res 1958;6(6):791–812.

[21] Johnson DS, McGeoch LA. The traveling salesman problem: a case study in local optimization. Local Search Comb Optim 1997;1:215–310.

[22] Kılcı F, Kara BY, Bozkaya B. Locating temporary shelter areas after an earthquake: a case for turkey. Eur J Operational Res 2015;243(1):323–32.

[23] Floyd RW. Algorithm 97: shortest path. Commun ACM 1962;5(6):345.

**Bahar Yetis Kara** is an Associate Professor in the Department of Industrial Engineering at Bilkent University. Dr. Kara holds an M.S. and Ph.D. degree from Bilkent University Industrial Engineering Department, and she worked as a Postdoctoral Researcher at McGill University in Canada. Dr. Kara was awarded with “Research Excellence in Ph.D. Studies” by INFORMS (Institute for Operations Research and Management Science) UPS-SOLA. In 2008, Dr. Kara was awarded with TUBA-GEBIP (National Young Researchers Career Development Grant) reward. She attended the World Economic Forum in China in 2009. With her research and projects, IAP (Inter Academy Panel) and TWAS (The Academy of Science for the Developing World) awarded her with IAP Young Researchers Grant. Dr. Kara was elected as an associate member of the Turkish Academy of Sciences in 2012. She has been acting as a reviewer for the top research journals within her field. Her current research interests include distribution logistics, humanitarian logistics, hub location and hub network design, and hazardous material logistics.