EURO J Comput Optim (2016) 4:403–445
DOI 10.1007/s13675-016-0063-1

ORIGINAL PAPER

# Solution methodologies for debris removal in disaster response

**Nihal Berktaş**[1] · **Bahar Yetiş Kara**[1] · **Oya Ekin Karaşan**[1]

Received: 27 April 2015 / Accepted: 22 January 2016 / Published online: 9 February 2016


**Abstract** During the disaster response phase of the emergency relief, the aim is to reduce loss of human life by reaching disaster affected areas with relief items as soon as possible. Debris caused by the disaster blocks the roads and prevents emergency aid teams to access the disaster affected regions. Deciding which roads to clean to transport relief items is crucial to diminish the negative impact of a disaster on human health. Despite the significance of the problem during response phase, in the literature debris removal is mostly studied in the recovery or the reconstruction phases of a disaster. The aim of this study is providing solution methodologies for debris removal problem in the response phase in which effective and fast relief routing is of utmost importance. In particular, debris removal activities on certain blocked arcs have to be scheduled to reach a set of critical nodes such as schools and hospitals. To this end, two mathematical models are developed with different objectives. The first model aims to minimize the total time spent to reach all the critical nodes whereas the second minimizes the weighted sum of visiting times where weights indicate the priorities of critical nodes. Since obtaining solutions quickly is important in the early post-disaster, heuristic algorithms are also proposed. Two data sets belonging to Kartal and Bakırköy districts of İstanbul are used to test the mathematical models and heuristics.

**Keywords** Debris removal · Emergency relief · Disaster management

**Mathematics Subject Classification** 90C11 · 90C90

✉ Nihal Berktaş
nihal.berktas@bilkent.edu.tr

Bahar Yetiş Kara
bkara@bilkent.edu.tr

Oya Ekin Karaşan
karasan@bilkent.edu.tr

[1] Department of Industrial Engineering, Bilkent University, Ankara, Turkey

# 1 Introduction

From 1990 to 2012, average number of disasters observed each year was 340; each resulting in 240 million victims on the average. Although the annual number of disasters seems to decrease in 2012 and 2013, as seen in Fig. 1, it is still above 300; causing thousands of lives and leaving millions of people without food, clean water, shelter and medical care (EM-DAT 2013). To reduce further loss of life, providing emergency relief items to the disaster affected people as soon as possible is of utmost importance. Planning and organizing the flow of these items is called relief logistics and it is more complex than business logistics due to the conditions created by the disaster (Sheu 2007).

Unfortunately, there are several examples in the recent history showing the unique challenges possessed by the relief logistics. One of them is the lack of capable resources to handle the situation or the lack of ability to activate the resources on time. During Haiti earthquake in 2010 the limited ramp space of the airport and lack of fuel prevented humanitarian flights from entering the country (Murphy 2010). Furthermore, the uncertainties about demand may result in wrong or excessive donations which complicate handling and storage operations. Damage in communication systems and other infrastructure such as roads increases the complexity and difficulty in logistics. Again in Haiti earthquake the port was damaged and could not handle large ships so delivery of the emergency aids transported via ships were needed to be planned carefully. Involvement of many parties to control these resources creates another challenge since they have to communicate and coordinate efficiently. This challenge was sadly

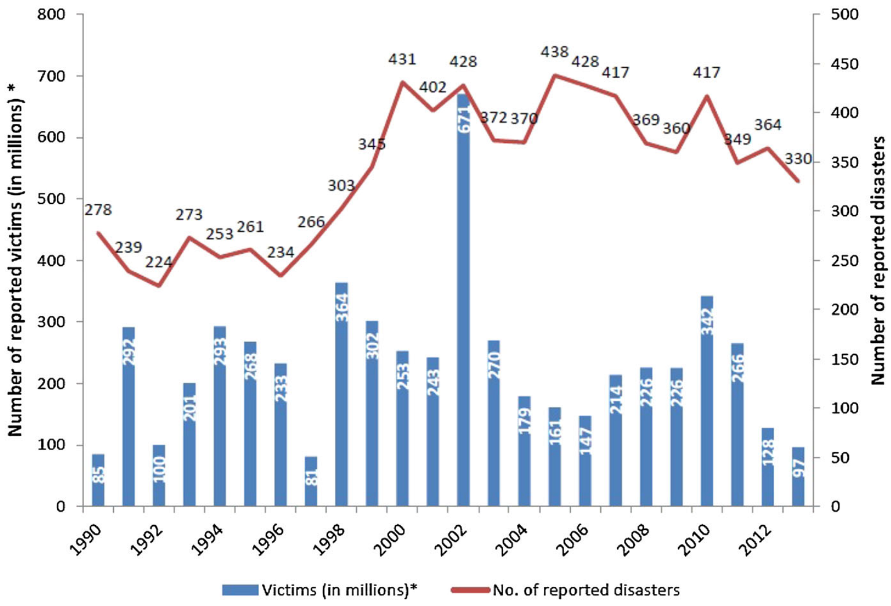


**Fig. 1** Disaster trends in terms of occurrence number and victims (sum of deaths and total affected) (EM-DAT 2013)

faced by Hurricane Mitch in 1998 when it took weeks for The International Federation of Red Cross and Red Crescent Societies (IFRC) to coordinate and distribute the donated reliefs (Samii et al. 2002).

When governments and institutions are not able to overcome these challenges, the effects of the disaster last for a long period of time, like in the example of Haiti earthquake; 98 % of the debris remained after 6 months from the earthquake and made the transportation impossible for the most part of the capital city (Lush 2010).

As these examples indicate, there are numerous factors to consider for relief logistics to be planned systematically and performed effectively. Transportation of the relief to the disaster affected people is one of, if not the most, significant aspects of relief logistics because in the absence of these items and services loss of lives increases drastically. There are many studies in the literature suggesting models to use in distribution and transportation of relief especially in the response phase of the disaster. As summarized in the recent reviews by Caunhye et al. (2012) and de la Torre et al. (2012) there are studies on relief transportation which incorporate different issues in their problems and solution methodologies. Hence the studies not only differ in terms of the operations research techniques but also in terms of the aspects of the problem approached. The most common issues considered in relief transportation models are vehicle characteristics such as number, type and capacity; number of depots, characteristics of demand and supply, travel time, i.e., deterministic or stochastic. To the best of our knowledge none of these models takes debris caused by the disaster into consideration.

Debris is caused by destruction of structures and vegetation and they block the roads and prevent accessibility to disaster affected areas. There are different type of debris; construction, vegetative, hazardous waste, properties such as white goods, vehicles, etc. (FEMA 2010). Hence debris differs from the normal waste in terms of content and amount as Hurricane Katrina proved it by producing more than 50 times the annual amount of daily solid waste in the USA in a few hours (Stephenson 2008). There are a few studies on debris removal in the literature and they are mostly focused on the recovery phase of the disaster aiming the complete removal and recycling of debris. However, debris becomes an obstacle in the response phase by blocking a road completely or partially and consequently complicates route determination in relief transportation. In this study, we focus on the debris removal operations in the response phase with relief routing incentives. Since the main goal is reaching disaster affected areas as soon as possible we consider debris removal not as complete clearance which takes months but as a sweeping operation so that wreckages are moved aside and enough space is created for relief carrying vehicles to pass.

The problem, Debris Removal in the Response Phase (DRR) is defined as reaching a set of predetermined critical disaster affected areas as soon as possible by traversing roads which may be blocked due to debris (Sahin et al. 2015). The concept of a predetermined set in this problem is the same as demand or delivery points in the relief transportation problems. The novelty of the problem is the consideration of the blocked roads and their usage at the expense of extra effort. This extra effort is expressed by the amount of time which is needed to sweep the debris and make the blocked arcs usable. Hence, the problem is a single vehicle routing problem which differs from TSP in two aspects: first, there is a subset of nodes that should be visited, and second, to use the blocked arcs the vehicle must spend some extra time on those arcs but only for their first usage. We can also interpret this extra time as a fixed cost defined for a subset of arcs. DRR is first defined by Sahin et al. (2015) who developed mathematical models and heuristics. One of the contributions of this paper is proposing new mathematical model for DRR using less variables and hence leading to computational efficiency. In DRR, each critical node is considered equally important and the aim is to minimize the total time to visit all critical nodes. However, this might not be realistic if the critical areas posses different characteristics in terms of population and level of urgency. Therefore, we define another problem, Prioritized Debris Removal in the Response Phase, (PDRR), in which the predetermined critical nodes have priorities and the objective is minimizing the total weighted visiting times of critical nodes.

We suggest mathematical models and heuristics for both problems. For the test of our solution methodologies, we use two data sets from provinces of Istanbul which faces a devastating earthquake almost in every century. Turkey is generally an earthquake prone country and statistics show that every 8 months a serious earthquake occurs in Turkey (Tübitak 2005). One of the recent destructive earthquakes is Van earthquake in 2011 which killed hundreds of people and injured thousands of them. The most catastrophic earthquakes in Turkey in the last century in terms of magnitude and casualties were Erzincan earthquake in 1939 and Marmara earthquake in 1999 which resulted in more than 17,000 casualties affecting more than a billion people. Even though our computational study has the scope of debris removal after an earthquake, our solution methodologies are general enough to be applied to other types of disasters such as tsunami, tornado and hurricane. We may need some adjustments in the assumptions and the parameters according to the type and place of the disaster, since the network features and debris characteristics depend on them. Our computational studies are based on residential districts of Istanbul and we assume that the debris can be moved aside using a vehicle, e.g, a bulldozer. The problem is considered as a single vehicle routing since each municipality has one such vehicle. As hurricanes can affect larger regions and cause a larger amount of debris, it would be more realistic to consider a multi-vehicle version of the problem with longer cleaning times for blocked roads.

In the next section we present a literature review on relief routing and debris removal in relief logistics. In Sect. 3, the first problem, DRR, is introduced and the MIP model and heuristics developed for this problem are described. In Sect. 4, we introduce our second problem, PDDR, for which another MIP model and heuristics are developed and presented. Section 5 describes the data sets and gives analyses of the computational results. Section 6 concludes with the general discussion of the study.

## 2 Literature review

With the increase in the number of studies in humanitarian logistics especially after 2000, various surveys are conducted. Altay and Green (2006) group the studies according to their place in the disaster timeline together with the solution methodologies and disaster type. Apte (2010) provides definitions, highlights the differences of humanitarian with other types of logistics while reviewing proposed models. The studies in the review by Caunhye et al. (2012) are categorized operationally as facility location, stock prepositioning, relief distribution and casualty transportation. de la Torre et al. (2012) focus on the research involving relief routing and classify them using the problem characteristics. Galindo and Batta (2013) review studies between 2005 and 2010 and compare them with the ones in Altay and Green (2006), then state and criticize the assumptions. Özdamar and Ertem (2014) provide a survey on the models developed for the response and recovery planning phases including the information system applications.

Since our study involves relief routing and debris removal, we focus on those studies in the literature. We first group the relief routing models according to the problem characteristics; whether it includes location decisions and casualty transportation or both. Then these models are examined in terms of more detailed features such as the fleet type, demand/supply, depots, commodities, etc. In the second part, we discuss research regarding debris removal which is mostly studied in the recovery phase.

### 2.1 Relief routing

One of the studies including relief routing with location and stock preposition is done by Chang et al. (2007) in flood emergency logistics. The problem is formulated as a two-stage stochastic programming problem where the first stage decisions are the locations of the local rescue bases. In the second stage, the number of equipments in each base, the allocation of the rescue teams and the flow of relief are decided. Mete and Zabinsky (2010) also propose a two-stage stochastic programming model of locating and distributing medical supplies. In their model, the second stage decision solely includes transportation, where locations and inventory levels are decided in the first one. They present a case study for earthquake scenarios in Seattle area. Duran et al. (2011) give a MIP formulation in which the decisions of which warehouse to open, the quantity of the commodities in the warehouses and the flows are jointly made. Rawls and Turnquist (2009) include uncertainty in both demand and network in their two-stage stochastic mixed integer program. Wang et al. (2014) use a nonlinear integer model which decides on the location of the distribution centers and routing of relief. They define reliabilities for each arc, which correspond to probability of successful traversal, and the reliability is included in the objective together with time and cost.

Afshar and Haghani (2012) develop a model for relief routing and locating temporary facilities following FEMA's logistic structure. Van Hentenryck et al. (2010) combine storage and routing decisions in their single commodity allocation problem defined for disaster recovery. The problem is to decide the set of depots to use, the amount of supply and the customer allocation to these depots. They use scenarios to model the stochasticity of the supply, demand and the infrastructure damage. Hence in the study damaged roads are taken into consideration by stochastic travel times.

Although the majority of the relief routing models focus solely on the routing decisions, still there are many studies which include casualty transportation. Most of these models are multi-period, deterministic and they both include resource allocation and flow decisions (Haghani and Oh 1996; Barbarosoglu and Arda 2004; Yi and Kumar 2007; Yi and Özdamar 2007).

Although the researchers' approach to the relief routing problem and the assumptions differ, there are some structures that are commonly used to model this problem. In the relief logistics literature, the majority of the researchers focus on multi-period and multi-commodity routing problems. Balçık and Beamon (2008) present such an MIP model with a tour-based formulation. The aim is to deliver relief items to demand points with minimum cost and unsatisfied demand. To incorporate the damaged roads in the model, they define the travel times according to the vehicle types so the travel time is taken very large for a specific type of vehicle if it is not possible for that type to use the road. In their multi period, multi-commodity relief distribution model, Tzeng et al. (2007) assume that the road condition is known but the relief is delivered only to the accessible areas. Lin et al. (2011) also propose a multi-period, multi-commodity MIP but with soft time windows. Their formulation is tour-based and for efficiency of the solution process the number of available tours are limited. They allow split delivery and they minimize the total time and unsatisfied demand together. Berkoune et al. (2012) define their relief routing problem similar to the one in Balçık and Beamon (2008). In addition to the weight and volume capacities they consider a limit on the traveling time per day. Sharif and Salari (2015) use a different approach to model the relief routing problem; they combine Open Vehicle Routing Problem with Covering Salesman Problem. There is heterogeneous set of vehicles that depart from a single depot and the demand is met either by visiting the node or by coverage. In addition, there are some studies which propose MIP models together with analyses on the effects of different objectives (Campbell et al. 2008; Huang et al. 2012).

From these studies, we observe that although the damage on roads are considered in some relief routing problems, using such roads at the expense of extra effort is not taken as an option. Next, we will examine the studies that mainly focus on debris removal and related road reconstruction studies.

### 2.2 Debris removal

Çelik et al. (2015) group the debris-related operations in disaster timeline where debris clearance is under response phase and debris collection is categorized as a recovery operation. This classification is due to difference in the priorities in those phases and their durations. In the studies on the debris collection, the aim is to remove the debris completely and properly dispose of it. Such a study is done by Fetter and Rakes (2012) who propose a facility location model which aims to maximize recycling with minimum cost. The model decides where to locate temporary disposal and storage reduction (TDSR) facilities among a set of possible locations. TDSRs may posses different technologies and they incur fixed and technological costs which are minimized together with the cost of collecting and transporting debris.

Hu and Sheu (2013) incorporate psychological effects of debris into the debris collection. They develop a multi-objective model which includes three conflicting costs: logistical, risk-induced and psychological. Logistical costs consist of operational costs related to transportation and recycling of debris. Risk-induced cost includes environmental risks associated with uncollected debris, storages and transportation. In psychological cost both disaster victims and people working in the recovery operations are considered.

The operation that we call debris removal or clearance is mostly referred to as road or network reconstruction or restoration in the literature. Chen and Tzeng (1999) study road construction in the recovery phase of a disaster. Feng and Wang (2003) develop a scheduling model for the response phase in which roads are repaired by work groups to maximize accessibility to the disaster affected people. Averbakh (2012) schedules the restoration of damaged edges after a disaster. There are multiple servers that repair roads to reach nodes. The time that the nodes are reached are called recovery times and the objective is to minimize these times. The unrealistic assumption in the model is negligence of the travel time of servers in the restored roads. Another network construction problem is studied by Averbakh and Pereira (2012) where the aim is to minimize the weighted total recovery time of all vertices. The recovery of a vertex is accomplished when it is connected the depot by the construction of the necessary edges. Both exact solution methods and fast heuristics are proposed in the article. In their latest study on network construction problem, Averbakh and Pereira (2015) consider vertices with due dates with two objectives: minimizing the maximum lateness and minimizing the number of tardy vertices. In all of these problems on network construction, the construction speed is taken incomparably slower than the travel speed.

Nurre et al. (2012) present a network design and scheduling problem to maximize the weighted sum of flow by deciding on which arcs to restore and add to the network. The restoration is done by work groups which need to be scheduled. Although the problem is multi-period the restoration does not necessarily go through all periods. The model can stop the restoration at any time since the aim is to maximize the total flow at the end of the restoration. Nurre and Sharkey (2014) propose a heuristic dispatching rule algorithm for the integrated network design and scheduling problem which has three main decisions: deciding the components of the network to improve, assigning the machines to these components and sequencing the processes on the machines which are parallel and identical. The performance of the network is measured by solving classical network optimization problems, i.e., maximum flow, minimum cost flow, shortest path and minimum spanning tree.

Aksu and Ozdamar (2014) focus on the first 3 days of the response phase and the problem is to gain accessibility to all locations as soon as possible by restoring links. Each node is required to be connected to a relief center and debris dump site; this connection is represented by access paths that are defined for each node. Hence the aim is to clean roads to make access paths clear in a given time. The objective maximizes the total weighted earliness of access paths' restoration times thus if a path is not clear it is not included in the objective.

The studies mentioned above make the assumption that the locations and the repair times of the damaged roads are known. Çelik et al. (2015) study the debris clearance problem where the location of the blocked arcs are known but the amount of resource required to clean a subset of blocked arcs is stochastic. The problem is multi-period and a schedule is made for the subsequent period in the beginning of each period. The objective is to maximize the total weighted flow sent to demand nodes by connecting the nodes to the supply with debris clearance. As in the study by Averbakh (2012) the travel times of the undamaged or cleaned arcs are neglected in this study, assuming that clearance times are very high compared to the travel times.

Pramudita et al. (2014) study location and routing problems of debris collection after disasters and use Location-Capacitated Vehicle Routing Problem (L-CVRP) formulation. Although the problem sounds similar to ours, the assumptions and the decisions taken by models make them distinct. In this study along with the routing, there is a location problem on the disposal sites and the allocation of the demands, blocked arcs, to this disposal sites. All blocked arcs have to be cleaned and it is assumed that the only path between two nodes is the shortest path. Furthermore, they define a matrix called access possibility to represent the condition of the arcs. The values of this matrix should be updated each time a required node is visited and it is referred to as a dynamic constraint but it is not clear how the matrix is updated in the model.

As previously stated in Sect. 1, Debris Removal in the Response Phase is first defined by Sahin et al. (2015). In the study a four indexed MIP model is suggested under the assumption that the blocked arcs and the time required to clean them are known. The model aims to minimize the total time spent to reach all the critical nodes. Construction and improvement heuristics are also developed for the problem. In the next section we describe the problem in detail and present our improved solution methodologies.

## 3 Debris removal in the response phase

### 3.1 Problem setting

Debris Removal in the Response Phase (DRR) is the problem of determining the route among the critical nodes and deciding on which roads to clean. Critical nodes, mostly stated as demand points in many studies, correspond to areas that need urgent relief. We consider these nodes as schools, hospitals and possible shelter areas. Debris caused by the disaster prevents accessibility to some roads which we call blocked arcs. We assume that the location of the blocked arcs is known and opening these arcs to traverse is a decision made by the model. The demands and supplies are not included in the model because the problem is considered just after the disaster in which even partial knowledge on these is not available. As stated by Apte (2010) this distinguishes disaster response from humanitarian relief during which there exists more information available on needs and less chaotic environment compared to response phase.

Therefore, in this phase we focus on making the critical locations reachable by routing a single vehicle among the critical nodes. For relief items and emergency aid teams to reach these critical locations there should be at least one path to those nodes from the supply node. By routing a single vehicle among all nodes, we make sure a debris-free route exits among them.

Let $G = (N, A)$ be a complete and symmetric graph where $N$ is the node set, partitioned into the critical nodes set $C$ and noncritical nodes set $NC$, and $A$ constitutes the arc set of the network. $s$ denotes the supply node and $s \in C$. Time required for traversing arc $(i, j) \in A$ is $t_{ij}$ and parameter $I_{ij}$ takes value 0 if the arc $(i, j)$ is blocked and is 1 otherwise. The effort spent on cleaning a blocked arc is measured in terms

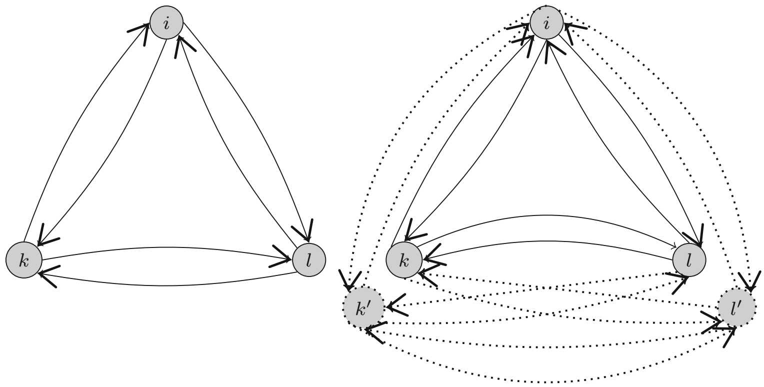


**Fig. 2** Original network $G = (N, A)$ (*left*), new network (*right*) $G' = (N', A')$ where *dotted nodes* and *arcs* are artificial

of time and it is denoted by $c_{ij}$ for arc $(i, j)$. Thus the time required to traverse a blocked arc $(i, j)$ for the first time is $t_{ij} + c_{ij}$. Since the network is symmetric if $(i, j)$ is blocked so is $(j, i)$ and removing debris on one of them makes both of them clean. Furthermore, it is assumed that an unblocked arc cannot be blocked again so for the subsequent usages of the arc only $t_{ij}$ amount of time is spent.

To be able to formulate the problem using three index variables, we duplicated the critical nodes and adjacent arcs. Hence, each critical node $k \in C$ has a duplicated version $k'$. These duplicated critical nodes are represented by set $C'$ and they are treated as noncritical nodes. Thus we have a new noncritical node set; $NC' = NC \cup C'$. The set of all nodes, $N'$, consist of the original critical set $C$ and the new noncritical set $NC'$. Since the adjacent arcs are also duplicated we define a new arc set $A' = A \cup \{(k', j), (j, k') : k' \in C', j \in N : j \neq k\} \cup (k', l') : k', l' \in C', k' \neq l'\}$.

In Fig. 2 above an example of duplication of nodes and arcs in a small network is illustrated. The nodes $k$ and $l$ are original critical nodes where node $i$ is an original noncritical node. The dotted nodes and arcs are the duplicated ones included in the new network. These artificial arcs have the same parameter values with the original ones so $t_{k'i} = t_{ki}$, $t_{k'l'} = t_{kl}$, etc. If the original arc $(k, i)$ is blocked then $(k', i)$ is also blocked and cleaning one of them makes all of them clean. We model the problem using this new graph $G' = (N', A')$. In the problem, the critical nodes are free to be used as intermediate nodes so the optimal route may require revisiting critical node $m$ while going from critical node $k$ to critical node $l$. To make this possible in the 3-indexed formulation, we create the duplicated critical nodes which act like noncritical ones. In the next subsection we give the formulation and explain the constraints together with the necessity of the creating this new graph in detail.

### 3.2 Proposed model

Under this setting, we formulate the problem as an MIP model which is a reformulation of the one in Sahin et al. (2015). The decision variables are as follows:

$y_{kl}$ = 1 if $l \in C$ is the first critical node visited after $k \in C$ (possibly with noncritical nodes in between) and 0 otherwise

$x_{ij}^k$ = 1 if $(i, j) \in A'$ is traversed while going to critical node $k$ from the previous critical node, and 0 otherwise

$C_k$ = time spent to reach critical node $k \in C \backslash s$ from the previous critical node (the time required for debris removal not included)

$B'_{ij}$ = 1 if $(i, j) \in A'$ is cleaned, and 0 otherwise

$B_{ij}$ = 1 if $(i, j) \in A$ is cleaned, and 0 otherwise

$p_k$ = time that node $k \in C$ is reached (debris removal not included)

TT = total travel time spent to visit all critical nodes (debris removal not included)

Before the formulation, we would like to note that the objective is to minimize the total time to visit all critical nodes. Thus, although the mathematical model implies that the vehicle returns to the supply node, the time spent on this return trip is not relevant and not included in the objective function.

This model minimizes the total time spent to visit all the critical nodes. The total time includes the regular traveling time and the time for the debris removal on the blocked arcs that are chosen to be used. Constraints (2) and (3) ensure that each critical node except the supply node has exactly one predecessor and successor critical node to form a visiting order. (4) guarantees that supply node is predecessor of exactly one of the critical nodes. These three assignment constraints construct a closed tour. Since (10) makes the visiting time of the supply node equal to zero, it ensures that the tour starts from the supply node.

(5), (6) and (7) jointly construct the paths among the critical nodes and they are the reason behind the duplication. If critical node $l$ is visited right after critical node $k$, (5) ensures that the vehicle leaves $k$ to go to $l$. The vehicle may go to the node $l$ directly or it can first go to a noncritical node $j \in NC'$. For the latter case, if vehicle visits an intermediate node $j$ to reach a critical node $k$, (6) ensures that the vehicle leaves this node; in other words, it makes the total flow entering $j$ and leaving $j$ equal. In this constraint, the first summation is over all nodes since the vehicle can depart from any node and the second sum is over the all noncritical nodes and the targeted critical node $k$. Constraint (7) guarantees that there is a flow entering each critical node.

$$\min \mathrm{TT} + \sum_{i,j \in N: i<j} c_{ij} B_{ij} \tag{1}$$

s.t.

$$\sum_{l \in C:\, l \neq k} y_{lk} = 1 \qquad \forall k \in C \backslash \{s\} \tag{2}$$

$$\sum_{l \in C:\, l \neq k} y_{kl} = 1 \qquad \forall k \in C \backslash \{s\} \tag{3}$$

$$\sum_{l \in C \setminus \{s\}} y_{sl} = 1 \tag{4}$$

$$\sum_{j \in NC' \cup \{l\}} x_{kj}^{l} = y_{kl} \qquad \forall k, l \in C\ k \neq l \tag{5}$$

$$\sum_{i \in N'} x_{ij}^{k} - \sum_{h \in NC' \cup \{k\}} x_{jh}^{k} = 0 \qquad \forall k \in C\ \forall j \in NC' \tag{6}$$

$$\sum_{i \in N'} x_{il}^{l} = 1 \qquad \forall l \in C \setminus \{s\} \tag{7}$$

$$C_l = \sum_{i,j \in N'} x_{ij}^{l} t_{ij} \qquad \forall l \in C \setminus \{s\} \tag{8}$$

$$p_l \geq p_k + C_l - (1 - y_{kl})M \qquad \forall k \in C, l \in C \setminus \{s\} \tag{9}$$

$$p_s = 0 \tag{10}$$

$$\mathrm{TT} \geq p_k \qquad \forall k \in C \tag{11}$$

$$y_{kl} + y_{lk} \leq 1 \qquad \forall k, l \in C, k \neq l \tag{12}$$

$$B'_{ij} \leq 1 - I_{ij} \qquad \forall i, j \in N' : i < j \tag{13}$$

$$\sum_{l \in C \setminus S} (x_{ij}^{l} + x_{ji}^{l}) \leq |C|(B'_{ij} + I_{ij}) \qquad \forall i, j \in N' : i < j \tag{14}$$

$$B'_{ij} + B'_{ij'} + B'_{i'j'} + B'_{ji'} \leq 4B_{ij} \qquad \forall i, j \in C, \forall i', j' \in C' \tag{15}$$

$$B'_{ij} + B'_{ij'} \leq 2B_{ij} \qquad \forall i \in NC, j \in C : i < j \tag{16}$$

$$B'_{ji} + B'_{ij'} \leq 2B_{ij} \qquad \forall i \in NC, j \in C : i > j \tag{17}$$

$$B'_{ij} \leq B_{ij} \qquad \forall i, j \in N : i < j \tag{18}$$

$$x_{ij}^{k} \in (0, 1) \qquad \forall i, j \in N',\ \forall k \in C \tag{19}$$

$$C_k \geq 0 \qquad \forall k \in C \tag{20}$$

$$B'_{ij} \in (0, 1) \qquad \forall i, j \in N' \tag{21}$$

$$B_{ij} \in (0, 1) \qquad \forall i, j \in N \tag{22}$$

$$y_{kl} \in (0, 1) \qquad \forall k, l \in C \tag{23}$$

$$p_k \geq 0 \qquad \forall k \in C \tag{24}$$

$$\mathrm{TT} \geq 0 \tag{25}$$

(8) calculates the travel time to go to critical node $l$ from the previous critical node. (9) assigns visiting times of critical nodes, excluding the time spent on debris removal and this constraint prevents sub-tours. (11) and the objective together force TT to be equal to the visiting time of the last visited critical node. (12) is a simple valid inequality that says either $k$ precedes $l$, $l$ precedes $k$ or they are not consecutively visited critical nodes.

(13) ensures that only a blocked arc can be cleaned. When an arc in set $A'$ is traversed, if the arc is blocked, the corresponding $B'$ variable becomes 1 due to constraint (14) and indicates that the arc is cleaned. If both an original arc, say $(k, j)$ and

an duplicated version of it, say $(k', j)$ are cleaned then both $B'_{k'j} = 1$ and $B'_{kj} = 1$. If we use variables $B'$ in the objective then the cleaning time would be counted twice. However, arcs $(k, j)$ and $(k', j)$ actually correspond to the same arc, thus the cleaning time should be added to the objective only once. This is guaranteed by the variable $B_{ij}$ which is defined for all $(i, j) \in A$. By constraints (15)–(18) we ensure that $B_{ij}$ is 1 if the vehicle uses the original blocked arc $(i, j)$ or one of its duplicated versions.

In the problem, as mentioned earlier the critical nodes are free to be used as intermediate nodes. If we do not duplicate the critical nodes, this formulation does not allow a second visit to any critical node. Suppose that the vehicle visits critical node $k$ for the first time and critical node $l$ is to be visited next so $y_{kl} = 1$. Then by constraint (5) $x^l_{kj} = 1$ for some node $j$. If the vehicle revisits critical node $k$, say to reach critical node $m$, then by constraint (6), which assures flow conservation, $x^m_{kj}$ must be 1 for some $j$. This is possible only if $y_{km} = 1$ and it cannot be since $y_{kl} = 1$. Hence a critical node should be visited only once if the original network is used in the formulation. By duplicating the critical nodes, we allow the vehicle to visit a critical node more than once because the duplicated nodes are treated as noncritical. Thus, while going to critical node $m$, the vehicle visits $k'$ which actually means it is revisiting critical node $k$.

Since in disaster response phase decisions must be taken quickly, we developed heuristics to get near optimal solutions faster, especially for the cases when the network is large.

### 3.3 Heuristic: routing with shortest paths

We develop a heuristic, Routing with Shortest Paths (RSP), which makes the route construction similar to the tour in the Traveling Salesman Problem (TSP). Since in TSP the vehicle visits each node, we focus on the critical nodes and assume that the vehicle goes from one critical node to the other directly. This direct way is the shortest path between these critical nodes. In other words, we fix the path that can be used between two critical nodes; to go to $l \in C$ from $k \in C$ the vehicle uses the shortest path between $k$ and $l$. Thus the problem becomes finding a visiting order among the critical nodes using the shortest path lengths.

In this approach, we need to pay attention that blocked arcs can be used in these shortest paths and their cleaning times should be included in the path lengths. Hence there is a possibility that two or more shortest paths use and clean the same blocked arc. Since these paths are calculated separately they both include the cleaning time of the blocked arc. Hence we need to modify the regular TSP formulation so that if there is blocked arc used in the chosen shortest paths its cleaning time is taken only once.

This solution method may not result in an optimal solution since it calculates shortest paths between each node pair separately and consequently overlook the benefit of clearing a blocked arc that does not lie on a shortest path between two critical nodes to reduce the total route time. There may exist a blocked arc $(i, j)$ which is not on any of the shortest paths, but the cleaning and using this arc more than once may give smaller route time than the one found using RSP.

First we apply Dijkstra's algorithm to find the shortest paths among the critical nodes. The travel time of arc $(i, j)$ is taken as $t_{ij} + c_{ij}(1 - I_{i,j})$ to include the cleaning times. Then an MIP model is solved to find the visiting order among the critical nodes. The formulation is based on the original network G. The arcs that are used in the shortest path between critical nodes $k$ and $l$ is listed in the set $P_{kl}$ for all $k, l \in C$ and the time of this path is denoted as $\text{time}_{kl}$. The decision variables are as follows:

$y_{kl}$ = 1 if $l \in C$ is the first critical node visited after $k \in C$ (possibly with noncritical nodes in between) and 0 otherwise

$x_{ij}$ = 1 if $(i, j) \in A$ is used, and 0 otherwise

$B_{ij}$ = 1 if $(i, j) \in A$ is cleaned, and 0 otherwise

$p_k$ = time that node $k \in C$ is reached (debris removal not included)

TT = total travel time spent to visit all critical nodes (debris removal not included)

$$\min \text{TT} + \sum_{i,j \in N: i<j} c_{ij} B_{ij} (1 - I_{ij}) \tag{26}$$

s.t.

$$\sum_{l \in C:\, l \neq k} y_{lk} = 1 \qquad \forall k \in C \tag{27}$$

$$\sum_{l \in C:\, l \neq k} y_{kl} = 1 \qquad \forall k \in C \tag{28}$$

$$y_{kl} \leq x_{ij} \qquad \forall k, l \in C\ l \neq s, \forall (i, j) \in P_{kl} \tag{29}$$

$$x_{ij} + x_{ji} \leq 2B_{ij} \qquad \forall i, j \in N \tag{30}$$

$$p_l \geq p_k + \text{time}_{kl} - (1 - y_{kl})M \qquad \forall k \in C, l \in C \backslash \{s\} \tag{31}$$

$$(10\text{–}11), (22\text{–}25)$$

$$x_{ij} \in (0, 1) \qquad \forall i, j \in N \tag{32}$$

First two constraints are the classical TSP assignment constraints among the critical nodes. (29) ensures that if $l$ is visited after $k$ then arcs in the shortest path between $k$ and $l$ are used. (30) implies that a blocked road must be cleaned if it is used. (31) assigns the visiting times of critical nodes. The total travel time and the cleaning time is minimized with the objective function. Here the variable TT corresponds to the total travel time spent to reach all critical nodes since it is maximum of $p_k$ variables and the cleaning times of the blocked arcs that are used in the shortest paths are added separately to the objective.

We tested the RSP heuristic and the MIP formulation for DRR using data sets which will be described in Sect. 5. These solution methodologies are compared in terms of solution times and the solution quality. Now we move to the next problem, namely, the Prioritized Debris Removal during Response Phase (PDRR).

## 4 Prioritized debris removal during response phase

### 4.1 Problem setting

Treating each critical node equally might not be realistic since the characteristics of the nodes differ. It is reasonable to give some nodes priority if they are highly populated or more vulnerable. When the amount of debris, the number of blocked arcs and the number of critical nodes are high, the time spent to reach all nodes may take hours. Therefore, considering the weights or priorities of critical nodes and reaching the higher weighted ones sooner increases the overall benefit. For that purpose we developed a second model which minimizes the weighted sum of visiting times. The same network, $G'$, is used to formulate the problem. The weights of the critical nodes are denoted by $w_k$ for $k \in C$ for this model.

### 4.2 Proposed model

In the DRR model the variable $p_k$ corresponds to the time that critical node $k$ is reached but only considering the travel times. To minimize the weighted visiting times actual visiting times are needed. Thus, if an arc $(i, j)$ is cleaned while going to critical node $k$, the time required for debris removal should be included in $p_k$. Therefore we need to know which arc is cleaned to reach a specific critical node. Since an arc remains open once it is cleaned, spending debris removal time on that arc in the next usages must be prevented as in the first problem. To ensure that we need to know whether a critical node is visited earlier or later than another critical node and guarantee that a blocked arc is cleaned on its first usage. This necessitates different set of decision variables. Below the decision variables used in this formulation are given and the new ones are indicated in bold.

$y_{kl} =$ 1 if $l \in C$ is the first critical node visited after $k \in C$ (possibly with noncritical nodes in between) and 0 otherwise

$x_{ij}^{k} =$ 1 if $(i, j) \in A'$ is traversed while going to critical node $k$ from the previous critical node

$C_k =$ time spent to reach critical node $k \in C \backslash s$ from the previous critical node (the time required for debris removal not included)

$\mathbf{a_{kl}} =$ **1 if $\mathbf{l \in C}$ is visited after $\mathbf{k \in C}$, and 0 otherwise**

$\mathbf{v_{ij}^{k}} =$ **1 if $\mathbf{(i, j) \in A'}$ is cleaned to reach node $\mathbf{k \in C}$, and 0 otherwise**

$\mathbf{r_k} =$ **time that node $\mathbf{k \in C}$ is reached**

The variable $a_{kl}$ is different than $y_{kl}$ since it takes value 1 if critical node $k$ is visited any time before critical node $l$, not only when they are consecutively visited. This variable is required to add the cleaning time of an arc to the right visiting time and it is explained in detail after the formulation. Moreover due to this variable, the vehicle no longer can form a closed tour so it starts from the supply node but does not return to it. Hence the path finishes when the last critical node is visited.

$$\min \sum_{k\in C\backslash\{s\}} w_k r_k \tag{33}$$

s.t.

(5–8), 12

$$\sum_{k\in C:\, k\neq l} y_{kl} = 1 \qquad \forall l \in C\backslash\{s\} \tag{34}$$

$$\sum_{k\in C, l\in C\backslash\{s\}:\, k\neq l} y_{kl} = |C\backslash\{s\}| \tag{35}$$

$$\sum_{l\in C:\, l\neq k} y_{kl} \leq 1 \qquad \forall k \in C \tag{36}$$

$$a_{kl} \geq y_{kl} \qquad \forall k, l \in C \tag{37}$$

$$a_{kl} + a_{lk} = 1 \qquad \forall k, l \in C, k \neq l \tag{38}$$

$$a_{ml} \geq a_{mk} + y_{kl} - 1 \qquad \forall k, l, m \in C, k \neq l \tag{39}$$

$$a_{sl} = 1 \qquad \forall l \in C\backslash\{s\} \tag{40}$$

$$r_s = 0 \tag{41}$$

$$r_l \geq r_k + C_l + \sum_{i,j\in N: i<j} v_{ij}^k c_{ij} + (1 - y_{kl})\, M \qquad \forall k \in C \;\; \forall l \in C\backslash\{s\} \tag{42}$$

$$\sum_{l\in C\backslash\{s\}} v_{ij}^l \leq 1 - I_{ij} \qquad \forall i, j \in N : i < j \tag{43}$$

$$2 - v_{ij}^l \geq x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{ji'}^k + x_{j'i}^k + x_{ij'}^k + x_{i'j'}^k + x_{j'i'}^k + a_{kl} \qquad \forall i, j, k, l \in C : i < j, I_{ij} = 0, k \neq l \tag{44}$$

$$2 - v_{ij}^l \geq x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{ji'}^k + a_{kl} \qquad \forall i, k, l \in C, j \in NC : i < j, I_{ij} = 0, k \neq l \tag{45}$$

$$2 - v_{ji}^l \geq x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{i'j}^k + a_{kl} \qquad \forall i, k, l \in C, j \in NC : i > j, I_{ij} = 0, k \neq l \tag{46}$$

$$2 - v_{ij}^l \geq x_{ij}^k + x_{ji}^k + a_{kl} \qquad \forall k, l \in C, i, j \in NC : i < j, I_{ij} = 0, k \neq l \tag{47}$$

$$|C\backslash\{s\}| \sum_{k\in C\backslash\{s\}} v_{ij}^k \geq \sum_{k\in C\backslash\{s\}} (x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{ji'}^k + x_{j'i}^k + x_{ij'}^k + x_{i'j'}^k + x_{j'i'}^k) \qquad \forall i, j \in C\backslash\{s\} : I_{ij} = 0, i < j \tag{48}$$

$$|C\backslash\{s\}| \sum_{k\in C\backslash\{s\}} v_{ij}^k \geq \sum_{k\in C\backslash\{s\}} (x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{ji'}^k) \qquad \forall i \in C\backslash\{s\}, j \in NC : I_{ij} = 0, i < j \tag{49}$$

$$|C\backslash\{s\}| \sum_{k\in C\backslash\{s\}} v_{ji}^k \geq \sum_{k\in C\backslash\{s\}} (x_{ij}^k + x_{ji}^k + x_{i'j}^k + x_{ji'}^k) \qquad \forall i \in C\backslash\{s\},\ j \in NC : I_{ij} = 0, i > j \tag{50}$$

$$|C\backslash\{s\}| \sum_{k\in C\backslash\{s\}} v_{ij}^k \geq \sum_{k\in C\backslash\{s\}} (x_{ij}^k + x_{ji}^k) \qquad \forall i, j \in NC : I_{ij} = 0, i < j \tag{51}$$

$$x_{ij}^k \in (0, 1) \qquad \forall i, j \in N',\ \forall k \in C \tag{52}$$

$$v_{ij}^k \in (0, 1) \qquad \forall i, j \in N,\ \forall k \in C \tag{53}$$

$$a_{kl}, y_{kl} \in (0, 1) \qquad \forall k, l \in C \tag{54}$$

$$r_k, C_k \geq 0 \qquad \forall k \in C \tag{55}$$

Constraint (34) ensures that each critical node except the supply node has a predecessor critical node. (35) limits the total number of assignments to the number of critical nodes that we need to reach. (36) implies that a critical node may have a successor critical node or not. These are different than the assignment constraints of the previous formulation because the vehicle does not return to the supply node. This is needed because of the variable $a_{kl}$ and constraint (38) which assures either $k \in C$ is visited before $l \in C$ or vice versa.

By (37) if $k \in C$ is visited just before $l \in C$ then $k$ is visited before $l$ and by (39) we satisfy that any critical node $m$ which is visited before $k$ is also visited before $l$. The supply node is guaranteed to be the start node with constraints (40) and (41). Constraint (42) eliminates sub-tours between critical nodes and assigns visiting times including the time spent on debris removal. Thus if an arc $(i, j)$ is cleaned while going to critical node $k$, its cleaning time $c_{ij}$ is added to $r_k$.

With (43) it is guaranteed that an arc is cleaned only once and only if it is blocked. Constraints (44)–(47) prevent a blocked arc from being cleaned in latter usage. For example if a blocked arc $(i, j)$ or one of its artificial versions have been traversed while going to critical node $k$ and if $k$ is visited before critical node $l$, then $(i, j)$ cannot be cleaned while going to critical node $l$. Constraints (48)–(51) ensure that a blocked arc is cleaned while going to a critical node to be traversed to reach any critical node. Hence a blocked arc $(i, j)$ is cleaned if it is used at least once. These last constraints (44)–(51) together guarantee that when a blocked arc is used, it is cleaned once and debris is removed on its first usage.

### 4.3 Heuristic: prioritized routing with shortest paths

For the problem Prioritized Debris Removal in the Response Phase, we develop a heuristic by applying the same approach in RSP heuristic. Again the shortest paths among all critical nodes are obtained by Dijkstra's algorithm and an MIP is formulated to minimize the weighted sum of visiting times. Hence the MIP model finds a visiting order among the critical nodes, using the shortest paths and takes the priorities of the nodes into consideration. The decision variables are as follows:

$y_{kl}$ = 1 if $l \in C$ is the first critical node visited after $k \in C$ (possibly with noncritical nodes in between) and 0 otherwise

$a_{kl}$ = 1 if $l \in C$ is visited after $k \in C$, and 0 otherwise

$x_{ij}^{kl}$ = 1 if $(i, j) \in A$ is traversed while going from critical node $k$ to critical node $l$

$v_{ij}^{kl}$ = 1 if $(i, j) \in A$ is cleaned while going from $k \in C$ to $l \in C$, and 0 otherwise

$r_k$ = time that node $k \in C$ is reached

In this problem, the exact visiting times of critical nodes are required as explained earlier. To find the time that the vehicle reaches critical node $l$, $r_l$, we need to know the visiting time of the previous node, $r_k$, and all arcs that are used and cleaned between the shortest path from $k$ to $l$. Instead of variables $x_{ij}$ and $B_{ij}$ that are used in RSP heuristic, we define variables $x_{ij}^{kl}$ and $v_{ij}^{kl}$ to identify which arcs are cleaned on the path $p_{kl}$.

Constraint (57) ensures that if $l$ is visited after $k$ then the arcs in the shortest path between $k$ and $l$ are used. (58) implies that if a critical node $m$ is visited on the path from $k$ to $l$ then $m$ must be visited before $k$ and $l$. (59) and (60) guarantee that arc $(i, j)$ can be cleaned while going from $k$ to $l$ if the arc $(i, j)$ is blocked and it is in the shortest path between $k$ and $l$. By constraints (61) and (62) we make sure that the arc is cleaned in its first usage; (61) ensures that the arc is not cleaned if it used before while (62) forces that the arc to be cleaned in some path. (63) assigns the visiting times to the critical nodes.

$$\min \sum_{k \in C \setminus \{s\}} w_k r_k \tag{56}$$

s.t.

(34–41)

$$y_{kl} = x_{ij}^{kl} \qquad \forall k, l \in C\ l \neq s, \forall (i, j) \in P_{kl} \tag{57}$$

$$a_{ml} \geq x_{im}^{kl} \qquad \forall k, l, m \in C\ l, m \neq s, \forall i \in N \tag{58}$$

$$v_{ij}^{kl} = 0 \qquad \forall k, l \in C, \forall i, j \in N : I_{ij} = 0 \tag{59}$$

$$v_{ij}^{kl} \leq x_{ij}^{kl} + x_{ji}^{kl} \qquad \forall k, l \in C, \forall i, j \in N : i < j \tag{60}$$

$$2 - v_{ij}^{kl} \geq x_{ij}^{mn} + x_{ji}^{mn} + a_{mk} \qquad \forall k, m \in C, \forall l, n \in C \setminus \{s\} : k \neq l, m \neq n \quad \forall i, j \in N : I_{ij} = 0, i < j \tag{61}$$

$$|C \setminus \{s\}| \sum_{\substack{k,l \in C \\ k \neq l, l \neq s}} v_{ij}^{kl} \geq \sum_{\substack{k,l \in C \\ k \neq l, l \neq s}} (x_{ij}^{kl} + x_{ji}^{kl}) \qquad \forall i, j \in N : I_{ij} = 0, i < j \tag{62}$$

$$r_l \geq r_k + \text{time}_{kl} - (1 - y_{kl})M + \sum_{\substack{i,j \in N, i<j \\ I_{i,j}=0}} c_{ij} v_{ij}^{kl} \qquad \forall k \in C, l \in C \setminus \{s\} \tag{63}$$

$$x_{ij}^{kl}, v_{ij}^{kl} \in (0, 1) \qquad \forall i, j \in N, \forall k, l \in C \tag{64}$$

$$a_{kl}, y_{kl} \in (0, 1) \qquad \forall k, l \in C \tag{65}$$

$$r_k \geq 0 \qquad \forall k \in C \tag{66}$$

In the next section we present the data sets which are used in the analyses of the exact and heuristic solution methodologies proposed for DRR and PDRR.

## 5 Computational study

### 5.1 Data sets

All solution methodologies are tested using different data sets based on two districts of İstanbul, Turkey. The first district, Kartal, has 45 nodes, seven of which are critical. The second district, Bakırköy, has 73 nodes including 15 critical nodes. Hereafter by critical nodes we refer to the critical nodes excluding the supply node. In the data sets, the critical nodes correspond to the neighborhoods which are close to schools and hospitals. Detailed information about these data sets can be found in the studies by Kılcı et al. (2015) and Sahin et al. (2015).

The maps in Figs. 3 and 4 show the locations of supply nodes and critical nodes in Kartal and Bakırköy, respectively. In both, stars represent the supply node, triangles correspond to schools and squares are the locations of the nodes near hospitals. The node numbers for critical nodes and supply node are also given in Table 1.

For the computational analyses we use the instances that are generated by Sahin et al. (2015). In their study, the travel times are obtained using the actual travel distance between nodes and taking the vehicle's speed as 20 km/h. The matrix is symmetric and satisfies the triangle inequality.

To create different scenarios Sahin et al. (2015) consider four levels of earthquake severity. With the higher the severity of earthquake (SOE), we have higher number of blocked arcs. The blocked arc ratios (BAR) taken for each level of severity are presented in Table 2.

For each severity level, using the BAR values five I matrices are generated randomly. Thus, for each level of severity there are five instances with different sets of blocked arcs. Since there are four levels of severity, there are 20 instances. Further-

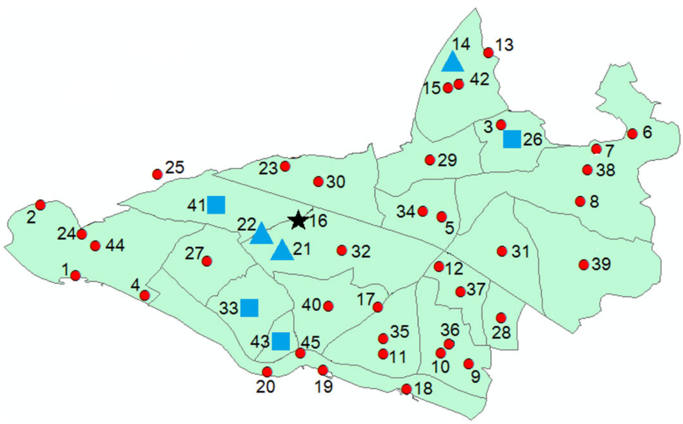


**Fig. 3** The locations of the supply node and the critical nodes in Kartal

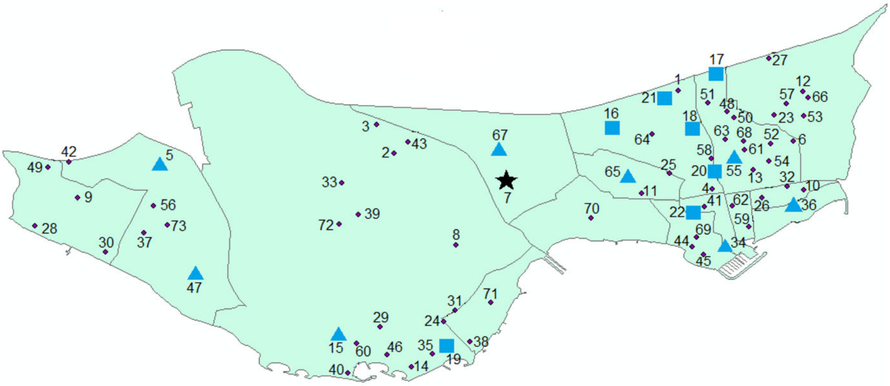


**Fig. 4** The locations of the supply node and the critical nodes in Bakırköy

**Table 1** Features of Kartal and Bakırköy graphs

| | Kartal | Bakırköy |
|---|---|---|
| # of nodes | 45 | 73 |
| Supply node | 16 | 7 |
| Total # of critical nodes | 7 | 15 |
| # of schools (school nodes) | 3 (14, 21, 22) | 8 (5, 15, 34, 36, 47, 55, 65, 67) |
| # of hospitals (hospital nodes) | 4 (26, 33, 41, 43) | 7 (16, 17, 18, 19, 20, 21, 22) |

**Table 2** Severity of earthquake, corresponding BAR values

| SOE | BAR (%) | BAR value for Kartal (%) | BAR value for Bakırköy (%) |
|---|---|---|---|
| 1 | 0–20 | 12.5 | 19 |
| 2 | 20–50 | 44.5 | 23 |
| 3 | 50–80 | 58 | 54 |
| 4 | 80–100 | 81.9 | 82 |

more, debris cleaning times on blocked arcs are calculated in two different ways, both depending on the severity of the earthquake and the travel time of the arc. The lower cleaning time is direct multiplication of the travel time with the SOE value since severe earthquakes cause higher amount of debris and the cleaning time is also dependent to the travel distance/time of that arc. The higher cleaning time is defined to include some randomness. $c_{ij}$ denotes the higher cleaning time and $c'_{ij}$ denotes the lower. They are calculated as follows:

$$c_{ij} = \mathrm{SOE} * t_{i,j} + U\left[0, \max_{(i,j)\in A} t_{ij}\right]$$

$$c'_{ij} = \mathrm{SOE} * t_{i,j}$$

**Table 3** Severity of earthquake, corresponding BAR values for the second Kartal set

| SOE | BAR values for the second Kartal set (%) |
|---|---|
| 1 | 40, 20, 0 |
| 2 | 50, 30, 10 |
| 3 | 70, 40, 20 |
| 4 | 80, 50, 30 |

Hence there are a total of 40 instances generated for Kartal data set; 20 with $c_{ij}$ (K1...K20) and 20 with $c'_{ij}$ (K1$'$ ...K20$'$). For example the locations of blocked arcs are different in K1, K2, K3, K4, K5 but the same cleaning effort is used for all. The only difference between K20 and K20$'$ is the cleaning times. These sets belonging to Kartal district are generated and used by Sahin et al. (2015). For Bakırköy data set, again 40 instances are generated in the same manner, but they are used with three different sets of critical nodes: only schools, only hospitals and both. Since Bakırköy is a larger district compared to Kartal, in addition to the original critical set, the computations are done by dividing the critical set into two.

For Kartal district, to consider some correlation among the blocked arcs we generate another data set. For this set a hypothetical disaster center is chosen and the blocked arc ratios are determined according to the closeness to this center. We take the center as node 16 which is supply node and close to the center of the district. Again we use four degrees of severity of earthquake (SOE) and for each of them three different BAR values are used to determine the number of blocked arcs. The first BAR value is used for the arcs whose adjacent nodes are in 7-min distance. The second is used for the ones that are $>7$ and $<15$, and the last one is used for the rest. These BAR values are given in Table 3. For each SOE, five different I matrices are generated and the lower cleaning times are used for these instances (K21...K40).

For the second model, to determine the weights assigned to each critical node, we use the population of neighborhoods to which each critical node belongs. The population of the neighborhoods are obtained from Turkish Statistical Institute and they are normalized so that each critical node has a weight value out of 100, and the sum of all weights is equal to 100. The normalization is done only for the nodes included in the set for the related instance. Thus while calculating the weights for Bakırköy instances in which the critical nodes are only schools, the hospitals are not considered.

In the following section we analyze the efficiency of the MIP models and heuristics with these instances.

### 5.2 Analyses on DRR

The experiments on the mathematical model and the heuristic (RSP) proposed for DDR are conducted using Java and CPLEX 12.6 on a 4XAMD Opteron Interlagos 16C 6282SE 2.6G 16M 6400MT computer. First we present the solutions obtained

**Table 4** DRR model test results with Kartal instances (higher cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\|C\backslash\{s\}\| = 7$ $c_{ij}$ | K1 | 44 | 39.07 | 0 |
| | K2 | 43 | 50.25 | 0 |
| | K3 | 44 | 51.13 | 0 |
| | K4 | 43 | 43.04 | 0 |
| | K5 | 43 | 42.63 | 0 |
| SOE = 2 $\|C\backslash\{s\}\| = 7$ $c_{ij}$ | K6 | 48 | 49.25 | 0 |
| | K7 | 50 | 56.34 | 0 |
| | K8 | 51 | 58.22 | 0 |
| | K9 | 49 | 68.92 | 0 |
| | K10 | 48 | 41.44 | 0 |
| SOE = 3 $\|C\backslash\{s\}\| = 7$ $c_{ij}$ | K11 | 53 | 71.55 | 0 |
| | K12 | 63 | 162.14 | 0 |
| | K13 | 68 | 114.85 | 0 |
| | K14 | 46 | 57.72 | 0 |
| | K15 | 47 | 58.17 | 0 |
| SOE = 4 $\|C\backslash\{s\}\| = 7$ $c_{ij}$ | K16 | 109 | 616.1 | 1 |
| | K17 | 82 | 372.72 | 0 |
| | K18 | 110 | 551.87 | 1 |
| | K19 | 90 | 348.14 | 1 |
| | K20 | 101 | 490.14 | 3 |

with Kartal instances by the MIP model and the heuristic and then the analyses on Bakırköy instances follow.

### *5.2.1 Kartal instances*

Parallel to our intuitions, as the severity of the earthquake increases, the CPU times increase as well. This is due to the increase in the number of blocked arcs and consequently the increase in the cleaning options. In other words, deciding which arcs to clean becomes more crucial when their number rises. Since the higher level of severity means higher cleaning times as well as more blocked arcs, the optimal values increase parallel to the earthquake impact. As seen in Tables 4 and 5 which represent Kartal results, when the severity of the earthquake is low, there is no cleaned arc in the optimal solutions. This is expected since the number of blocked arcs are low when the severity is low.

In the first five instances, either with higher and lower cleaning times, no arcs are cleaned and the optimal paths have only slight differences. For example in the solution of instance K2, depicted in Fig. 6, the vehicle goes from critical node 26 to critical node 14 directly. In instance K1 since arc 26–14 is blocked, the vehicle uses node 15 as an intermediate node between 26 and 14, as seen in Fig. 5. This change in the path

**Table 5** DRR model test results with Kartal instances (lower cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K1′ | 44 | 48.37 | 0 |
| | K2′ | 43 | 45.05 | 0 |
| | K3′ | 44 | 43.73 | 0 |
| | K4′ | 43 | 51.62 | 0 |
| | K5′ | 43 | 51.45 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K6′ | 48 | 55.27 | 0 |
| | K7′ | 49 | 67.33 | 1 |
| | K8′ | 51 | 70.75 | 0 |
| | K9′ | 49 | 79.59 | 0 |
| | K10′ | 48 | 66.77 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K11′ | 51 | 91.29 | 1 |
| | K12′ | 63 | 125.43 | 0 |
| | K13′ | 67 | 145.93 | 1 |
| | K14′ | 46 | 66.84 | 0 |
| | K15′ | 47 | 62.44 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K16′ | 97 | 488.17 | 3 |
| | K17′ | 78 | 119.8 | 1 |
| | K18′ | 95 | 513.36 | 3 |
| | K19′ | 81 | 120.31 | 2 |
| | K20′ | 80 | 373.64 | 3 |

increases the total time only by one unit as it can be seen in the optimal values of instances K1 and K2 in Table 4. Similar optimal paths are observed for instance K3, K4 and K5 also. Hence, the optimal routes, route times and CPU times do not differ much among the instances when the earthquake severity is low.

As the severity increases, the location of the blocked arcs result in larger differences in the optimal solution values and CPU times. For example, instances K17 and K20 have the same number of blocked arcs, however while the K17's path time is 82 with no cleaning, K20's path time is 101 with three cleaned arcs. The same comments are valid when the cleaning times are lower. As seen Fig. 7, in instance K17 the vehicle visits intermediate nodes and prefer not to do any cleaning whereas in the optimal path for K20, the arcs 22–41, 22–21 and 33–43 are cleared from debris. These arcs are indicated by dotted arrows in Fig. 8.

Comparing the solutions of instances with high and low cleaning times, we see that with lower times, the cleaning starts sooner in terms of earthquake severity. Furthermore, the number of cleaned arcs are a bit higher when the cleaning times are low. For example, instances K16 and K16′ have the same severity level, the same number and the same location of blocked arcs, the only difference is the time required to clean blocked arcs. This difference increases the number of blocked arcs from 1 to 3 between

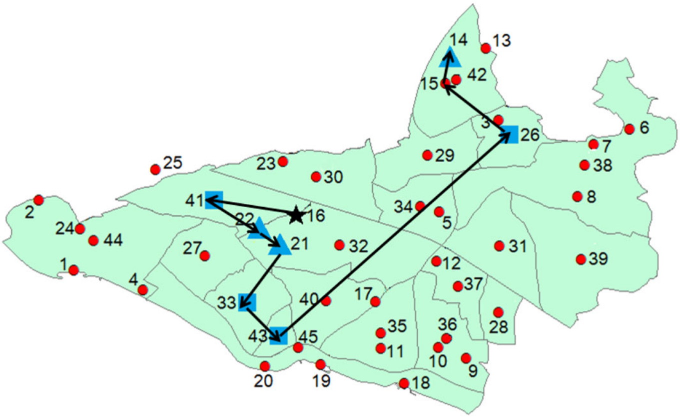


**Fig. 5** Optimal route of K1

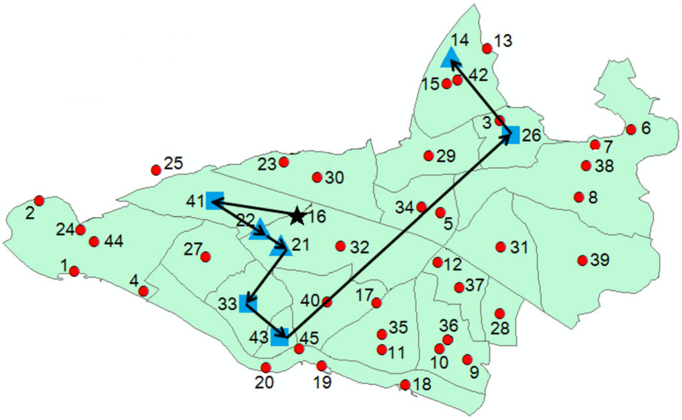


**Fig. 6** Optimal route of K2

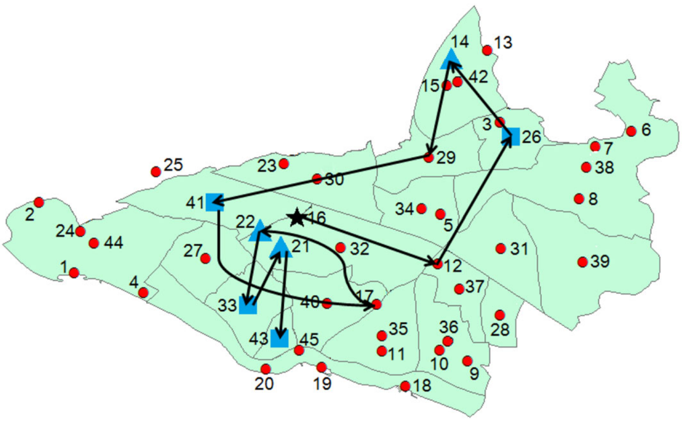


**Fig. 7** Optimal route of K17

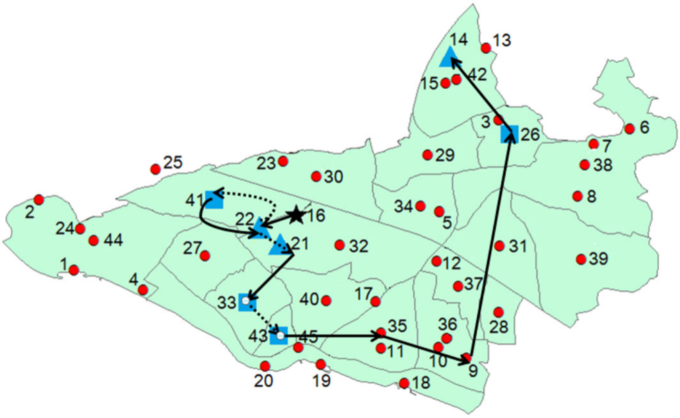


**Fig. 8** Optimal route of K20

**Table 6** CPU times of models proposed for DRR (in seconds)

| Instances | DDR model by Sahin et al. (2015) | | | DDR model by Berktaş et al. | | |
|---|---|---|---|---|---|---|
| | Min | Avg | Max | Min | Avg | Max |
| K1 ... K20 | 178.4 | 1441.73 | 9136.87 | 39.07 | 167.18 | 616.1 |
| K1′ ... K20′ | 152.9 | 1025.63 | 4864.74 | 43.73 | 134.36 | 513.36 |

K16 and K16′ while the total time drops from 109 to 97. The effect of cleaning time can also be observed by comparing instances K18 and K18′. When the cleaning is lower the optimal value decreases from 110 to 95 while the number of blocked arcs increase from 1 to 3.

When CPU times are examined, it is observed that on the average the solution times of the instances with lower cleaning times are less than the ones with higher cleaning times. The difference is more pronounced when the severity of the earthquake is higher. For example, the time needed to reach optimal solution for K18′ is less than half of the time needed for K18.

Since Sahin et al. (2015) test their model with the same instances we are able to compare the solution times of our model with theirs. The performances of the models are summarized in Table 6. It can be seen that for all instances from Kartal, same optimal solutions are reached eight times faster on the average by our model.

Next we present the results on the performance of the heuristic algorithm in terms of the solution quality for Kartal instances. For the instances when the cleaning time is higher, all instances are solved to optimality as seen in Table 7. With lower cleaning times we obtain optimal solutions in 15 out of 20 instances and the highest gap is 4.08 %. All Kartal instances are solved in <2 s.

The results of the DRR model and the RSP heuristic are summarized in the Table 8 using the second data set generated for Kartal district. For these instances, similar to the previous comments, we can say that as the severity of earthquake increases the average CPU time for the DRR model increase as well. However, the relationship

**Table 7** Performance of RSP heuristic on Kartal instances with higher and lower cleaning times

| Ins# | Opt. value | Heuristic | | | Ins# | Opt. value | Heuristic | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Value | Gap % | CPU (s) | | | Value | Gap % | CPU (s) |
| K1 | 44 | 44 | 0.00 | 1.57 | K1′ | 44 | 44 | 0.00 | 1.5 |
| K2 | 43 | 43 | 0.00 | 1.09 | K2′ | 43 | 43 | 0.00 | 1.28 |
| K3 | 44 | 44 | 0.00 | 1.33 | K3′ | 44 | 44 | 0.00 | 1.12 |
| K4 | 43 | 43 | 0.00 | 1.22 | K4′ | 43 | 43 | 0.00 | 1.18 |
| K5 | 43 | 43 | 0.00 | 1.05 | K5′ | 43 | 43 | 0.00 | 1.17 |
| K6 | 48 | 48 | 0.00 | 0.99 | K6′ | 48 | 48 | 0.00 | 1.21 |
| K7 | 50 | 50 | 0.00 | 0.98 | K7′ | 49 | 49 | 0.00 | 1.07 |
| K8 | 51 | 51 | 0.00 | 1.82 | K8′ | 51 | 51 | 0.00 | 1.25 |
| K9 | 49 | 49 | 0.00 | 1.13 | K9′ | 49 | 51 | 4.08 | 1.94 |
| K10 | 48 | 48 | 0.00 | 1.56 | K10′ | 48 | 48 | 0.00 | 1.17 |
| K11 | 53 | 53 | 0.00 | 1.02 | K11′ | 51 | 51 | 0.00 | 0.68 |
| K12 | 63 | 63 | 0.00 | 1.29 | K12′ | 63 | 63 | 0.00 | 1.23 |
| K13 | 68 | 68 | 0.00 | 1.20 | K13′ | 67 | 68 | 1.49 | 1.02 |
| K14 | 46 | 46 | 0.00 | 0.95 | K14′ | 46 | 46 | 0.00 | 1.06 |
| K15 | 47 | 47 | 0.00 | 0.96 | K15′ | 47 | 47 | 0.00 | 0.94 |
| K16 | 109 | 109 | 0.00 | 1.11 | K16′ | 97 | 98 | 1.03 | 1.43 |
| K17 | 82 | 82 | 0.00 | 1.05 | K17′ | 78 | 78 | 0.00 | 1.21 |
| K18 | 110 | 110 | 0.00 | 1.16 | K18′ | 95 | 97 | 2.10 | 0.92 |
| K19 | 90 | 90 | 0.00 | 1.16 | K19′ | 81 | 82 | 1.23 | 1.04 |
| K20 | 101 | 101 | 0.00 | 1.28 | K20′ | 80 | 80 | 0.00 | 1.01 |

between the SOE and the number of cleaned arcs in the previous results is not observed here. As seen in Tables 4 and 5, cleaning is observed mostly when the SOE is 3 and 4. In this data set, the number of blocked arcs are higher when close to the center so even if the SOE is low, higher number of blocked arc forces the model to clean some arcs. As given in Table 8, the RSP heuristic solves all instances to optimality in <2 s.

Next section continues with the results obtained from the Bakırköy instances.

### *5.2.2 Bakırköy instances*

The analyses on Bakırköy instances with DRR model lead to similar conclusions. When the severity of the earthquake increases, and consequently the number of blocked arcs and time required for debris removal increase, CPU times and optimal values rise. As seen in Tables 9, 10, 11 and 12 the amount of increase in the optimal values and CPU times vary according to the number of critical nodes and their locations. For example when the severity rises from 1 to 4, the increase in the objective function is higher for the instances where the critical node set consists of schools compared to the instances where only hospitals are critical.

**Table 8** DRR model and RSP heuristic test results with Kartal instances from the second data set

| Instance features | Ins. # | Optimal value | DRR CPU (s) | # of cleaned arcs | RSP Soln. | RSP CPU (s) |
|---|---|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K21 | 47 | 72.14 | 0 | 47 | 1.65 |
| | K22 | 46 | 47.25 | 0 | 46 | 1.18 |
| | K23 | 47 | 55.35 | 2 | 47 | 1.30 |
| | K24 | 46 | 60.15 | 1 | 46 | 1.45 |
| | K25 | 45 | 47.91 | 0 | 45 | 1.25 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K26 | 48 | 64.30 | 0 | 48 | 1.17 |
| | K27 | 53 | 90.07 | 1 | 53 | 1.02 |
| | K28 | 46 | 47.29 | 0 | 46 | 1.23 |
| | K29 | 47 | 82.27 | 0 | 47 | 1.97 |
| | K30 | 48 | 66.69 | 0 | 48 | 1.85 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K31 | 51 | 83.18 | 0 | 51 | 1.23 |
| | K32 | 61 | 116.04 | 1 | 61 | 0.90 |
| | K33 | 53 | 68.55 | 0 | 53 | 1.00 |
| | K34 | 55 | 81.69 | 0 | 55 | 1.08 |
| | K35 | 70 | 123.89 | 2 | 70 | 1.61 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K36 | 84 | 453.87 | 1 | 84 | 0.96 |
| | K37 | 50 | 70.49 | 0 | 50 | 1.34 |
| | K38 | 71 | 123.23 | 2 | 71 | 1.34 |
| | K39 | 66 | 158.06 | 2 | 66 | 0.94 |
| | K40 | 62 | 87.49 | 2 | 62 | 1.14 |

**Table 9** DRR model test results with Bakırköy instances (hospitals–higher cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B1 | 41 | 242.55 | 0 |
| | B2 | 38 | 215.5 | 0 |
| | B3 | 39 | 235.79 | 0 |
| | B4 | 40 | 293.67 | 0 |
| | B5 | 40 | 338.96 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B6 | 39 | 207.74 | 0 |
| | B7 | 39 | 165.41 | 0 |
| | B8 | 40 | 197.05 | 0 |
| | B9 | 42 | 294.85 | 0 |
| | B10 | 38 | 173.02 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B11 | 39 | 136.22 | 0 |
| | B12 | 42 | 190.2 | 0 |
| | B13 | 46 | 206.05 | 0 |
| | B14 | 48 | 203.78 | 0 |
| | B15 | 43 | 172.18 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B16 | 61 | 1197.11 | 0 |
| | B17 | 58 | 1437.28 | 0 |
| | B18 | 51 | 211.69 | 0 |
| | B19 | 59 | 329.28 | 0 |
| | B20 | 52 | 208.13 | 0 |

When we compare the instances concerned with the schools and hospitals, we see that the schools instances result in higher CPU times. For example, with instance B1 the optimal is reached in 454.17 s for schools and in 242.55 s for hospital as seen in Tables 9 and 11. This difference in the CPU times increases when SOE becomes 4 as in the instance B16–B20. The average CPU for these instances where SOE equals to 4 is 4058 s for schools and 677 s for hospitals. Therefore the location of the critical nodes has a significant effect on the CPU times.

Although lower cleaning time result in lower CPU times for Kartal instances, this observation cannot be generalized for the Bakırköy instances. For example, from Tables 11 and 12 we see that when the critical set only includes schools, CPU times of B9 and B10 are lower than B9′ and B10′, however B6, B7 and B8 are greater than B6′, B7′ and B8′, respectively. Furthermore not just the number of critical nodes but

**Table 10** DRR model test results with Bakırköy instances (hospitals–lower cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B1′ | 41 | 297.92 | 0 |
| | B2′ | 38 | 214.37 | 0 |
| | B3′ | 39 | 245.09 | 0 |
| | B4′ | 40 | 238.23 | 0 |
| | B5′ | 40 | 224.49 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B6′ | 39 | 189.75 | 0 |
| | B7′ | 39 | 180.87 | 0 |
| | B8′ | 40 | 349.11 | 0 |
| | B9′ | 42 | 239.42 | 0 |
| | B′10 | 38 | 218.22 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B11′ | 39 | 227.11 | 0 |
| | B12′ | 42 | 203.41 | 0 |
| | B13′ | 46 | 239.16 | 0 |
| | B14′ | 48 | 291.72 | 0 |
| | B15′ | 43 | 252.80 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B16′ | 59 | 444.2 | 0 |
| | B17′ | 57 | 497.27 | 0 |
| | B18′ | 51 | 375.97 | 0 |
| | B19′ | 55 | 1189.86 | 1 |
| | B20′ | 52 | 393.92 | 0 |

the size and features of the network have an effect on CPU times. When we look at the instance of Kartal and Bakırköy with seven critical nodes, namely instances K1–K5 and B1–B5 for hospitals, we see that the average CPU time of these Bakırköy instances is four times those of Kartal.

The heuristic, RSP, which is developed for DRR problem solves all Bakırköy instances to optimality when the critical node set consists of only hospitals. The average solution time is around 1.35 s. When the critical node set is only schools, RSP finds the optimal solution for all instances with higher cleaning time. When the cleaning times are lower, 16 instances out of 20 are solved to optimality; the gaps for instances B11′, B12′, B16′, B18′ are 5.97, 1.35, 4.6 and 1.92 %, respectively.

When hospitals and schools are both taken as critical, the number of critical nodes becomes 15 and reaching optimal solutions in a reasonable amount of time becomes

**Table 11** DRR model test results with Bakırköy instances (schools–higher cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\|C\backslash\{s\}\| = 8$ (schools) $c_{ij}$ | B1 | 52 | 454.17 | 0 |
| | B2 | 61 | 766.38 | 1 |
| | B3 | 52 | 463.86 | 0 |
| | B4 | 54 | 528.97 | 0 |
| | B5 | 52 | 480.68 | 0 |
| SOE = 2 $\|C\backslash\{s\}\| = 8$ (schools) $c_{ij}$ | B6 | 52 | 1389.95 | 0 |
| | B7 | 60 | 2549.11 | 0 |
| | B8 | 52 | 1050.84 | 0 |
| | B9 | 60 | 1877.6 | 0 |
| | B10 | 52 | 425.77 | 0 |
| SOE = 3 $\|C\backslash\{s\}\| = 8$ (schools) $c_{ij}$ | B11 | 71 | 1439.26 | 0 |
| | B12 | 80 | 2770.89 | 0 |
| | B13 | 74 | 1038.03 | 0 |
| | B14 | 71 | 731.99 | 0 |
| | B15 | 77 | 2460.8 | 0 |
| SOE = 4 $\|C\backslash\{s\}\| = 8$ (schools) $c_{ij}$ | B16 | 96 | 5271.01 | 0 |
| | B17 | 78 | 1971.88 | 0 |
| | B18 | 112 | 6974.84 | 0 |
| | B19 | 84 | 2974.57 | 0 |
| | B20 | 87 | 3098.48 | 0 |

extremely difficult. Nevertheless, to compare the performance of the DRR model and the RSP heuristic, we obtain the solutions by setting time limits to 2 h for the MIP model and for the heuristic we look at the solutions found in 5 min and 2 h. As seen in Tables 13 and 14 the heuristic outperforms the MIP model but we are not able to comment on the quality of the solutions since the optimal solutions are not available. The gaps show the percentage improvement over the DRR solution by the RSP heuristic. In Table 14 we have one negative gap which means the model finds better solution than the heuristic. When we compare the solutions reported by RSP heuristic within 5 min and 2 h, we see that few improved solutions are found after 5 min. Therefore, we suggest running the heuristic for 5 min to reach a feasible solution faster.

**Table 12** DRR model test results with Bakırköy instances (schools–lower cleaning times)

| Instances features | Instance # | Optimal value | CPU (s) (Cplex) | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B1′ | 52 | 775.45 | 0 |
| | B2′ | 58 | 517.7 | 1 |
| | B3′ | 52 | 563.33 | 0 |
| | B4′ | 54 | 357.18 | 0 |
| | B5′ | 52 | 883.73 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B6′ | 52 | 501.47 | 0 |
| | B7′ | 56 | 2432.34 | 1 |
| | B8′ | 52 | 437.61 | 0 |
| | B9′ | 56 | 2175.61 | 1 |
| | B′10 | 52 | 1502.76 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B11′ | 67 | 3112.28 | 1 |
| | B12′ | 74 | 3456.95 | 2 |
| | B13′ | 73 | 2717.89 | 1 |
| | B14′ | 69 | 1747.06 | 1 |
| | B15′ | 75 | 1128.25 | 1 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B16′ | 87 | 5118.85 | 3 |
| | B17′ | 78 | 2633.39 | – |
| | B18′ | 104 | 6676.95 | 4 |
| | B19′ | 79 | 3649.5 | 1 |
| | B20′ | 83 | 2826.31 | 2 |

### 5.3 Analyses on PDRR

For the second model the same data sets and instances are used by assigning appropriate weights to the critical nodes as explained in Sect. 5.1. Our preliminary analyses show that for the second model Gurobi 5 gives better CPU times than Cplex 12.6. Both solvers are tested under different MIP emphasis settings and the best results are obtained in the default setting for both solvers. The instance K1–K20 are solved for 2 h using both solvers. Out of 20 instances, Cplex 12.6 finds 18 optimal solutions in 2 h while Gurobi 5 reaches all optimal solutions with an average CPU time of 683 s. Therefore Gurobi 5 is used for the test of the second model using Java on the same computer.

**Table 13** Solutions of Bakırköy Instances by DRR model and RSP heuristic (all critical nodes, higher cleaning times)

| Ins # | Best obj by DRR model | Best obj by RSP heuristic (5 min) | Gap[a] (5 min) | Best obj by RSP heuristic (2 h) | Gap[a] (2 h) |
|---|---|---|---|---|---|
| B1 | 96 | 73 | 23.96 | 73 | 23.96 |
| B2 | 84 | 80 | 4.76 | 80 | 4.76 |
| B3 | 77 | 72 | 6.49 | 72 | 6.49 |
| B4 | 79 | 71 | 10.13 | 71 | 10.13 |
| B5 | 78 | 72 | 7.69 | 71 | 8.97 |
| B6 | 81 | 80 | 1.23 | 72 | 11.11 |
| B7 | 87 | 80 | 8.05 | 80 | 8.05 |
| B8 | 78 | 73 | 6.41 | 72 | 7.69 |
| B9 | 81 | 78 | 3.70 | 78 | 3.70 |
| B10 | 79 | 70 | 11.39 | 70 | 11.39 |
| B11 | 110 | 93 | 15.45 | 93 | 15.45 |
| B12 | 109 | 103 | 5.50 | 103 | 5.50 |
| B13 | 101 | 94 | 6.93 | 94 | 6.93 |
| B14 | 99 | 95 | 4.04 | 95 | 4.04 |
| B15 | 101 | 97 | 3.96 | 97 | 3.76 |
| B16 | 136 | 123 | 9.56 | 123 | 9.56 |
| B17 | 146 | 120 | 17.81 | 120 | 17.81 |
| B18 | 154 | 143 | 7.14 | 143 | 7.14 |
| B19 | 120 | 112 | 6.67 | 112 | 6.67 |
| B20 | 128 | 113 | 11.72 | 113 | 11.72 |

Gap = (DRR solution–RSP solution)/DRR solution × 100
[a] Gap indicates the improvement over DRR model solution

### 5.3.1 Kartal instances

The test results of the PDRR model on Kartal instances with higher and lower cleaning efforts are presented in Tables 15 and 16 respectively. From these tables we make similar deductions as in those for the previous problem. We see that debris removal occurs when the severity of the earthquake is higher. When the cleaning effort is high, blocked arcs are cleaned only when SOE equals to 4. When the cleaning effort is low, more blocked arcs are cleaned. For example in the solution of instance K19 only one arc is cleaned where three arcs are cleaned in K19′. Furthermore, similar to the previous results, when SOE increases, the CPU times and optimal values increase as well as seen in the test of the second model on Kartal instances.

When we compare the CPU times of both mathematical models using Kartal instances, Tables 4 and 15, we observe that solution times of PDRR model is much higher than those of DRR model. This is expected since we increase the number of variables and constraints to be able to find the exact visiting times of the critical nodes. In DRR model we only keep track whether the blocked arcs are cleaned or not.

**Table 14** Solutions of Bakırköy instances by DRR model and RSP heuristic (all critical nodes, lower cleaning times)

| Ins # | Best obj by DRR model | Best obj by RSP heuristic (5 min) | Gap[a] (5 min) | Best obj by RSP heuristic (2 h) | Gap[a] (2 h) |
|---|---|---|---|---|---|
| B1′ | 76 | 72 | 5.26 | 72 | 5.26 |
| B2′ | 78 | 78 | 0.00 | 74 | 5.13 |
| B3′ | 74 | 72 | 2.70 | 72 | 2.70 |
| B4′ | 73 | 71 | 2.74 | 71 | 2.74 |
| B5′ | 74 | 71 | 4.05 | 71 | 4.05 |
| B6′ | 82 | 72 | 12.20 | 72 | 12.20 |
| B7′ | 79 | 77 | 2.53 | 76 | 3.78 |
| B8′ | 90 | 72 | 20.00 | 72 | 20.00 |
| B9′ | 109 | 74 | 32.11 | 74 | 32.11 |
| B10′ | 80 | 70 | 12.50 | 70 | 12.50 |
| B11′ | 97 | 92 | 5.15 | 91 | 6.59 |
| B12′ | 97 | 97 | 0.00 | 97 | 0.00 |
| B13′ | 96 | 94 | 2.08 | 94 | 2.08 |
| B14′ | 104 | 93 | 10.58 | 93 | 10.58 |
| B15′ | 95 | 95 | 0.00 | 95 | 0.00 |
| B16′ | 115 | 119 | −3.48 | 119 | −3.48 |
| B17′ | 122 | 114 | 6.56 | 111 | 9.02 |
| B18′ | 145 | 139 | 4.14 | 139 | 4.14 |
| B19′ | 114 | 111 | 2.63 | 109 | 4.39 |
| B20′ | 114 | 110 | 3.51 | 110 | 3.51 |

Gap = (DRR solution − RSP solution)/DRR solution × 100

[a] Gap indicates the improvement over DRR model solution

In PDRR model, we are required to know when the arc is cleaned; going from which critical node to which critical node so that the cleaning time is added to the visiting time of the latter. The total route times of the optimal solutions found by both models are presented in Table 17. From this table we see that for each level of SOE some values are the same with the first model and some are higher. This implies that the change in the route and the total route time in the weighted case, significantly depends on the locations of the blocked arcs together with the weights. For example, the only difference in instance K11 and K13 are the location of the blocked arcs but for the first one both of the models reach optimal solutions with total route time equal to 53 while for instance K13, the total route times are 68 and 72 for the first and the second problem, respectively.

When we look at the instances K1 and K3, from Table 17 we see that the total route times are different; 44 and 47 respectively. However, from Table 15 it can be seen that the optimal values for the weighted case for these instances are the same which is 2035. This shows that same objective value for the second model does not necessarily imply that the optimal solutions have the same route and/or total route time.

**Table 15** PDDR model test results with Kartal instances (higher cleaning times)

| Instance features | Instance # | Optimal value | CPU time (Gurobi) | # of cleaned arcs | CPU time (Cplex) |
|---|---|---|---|---|---|
| SOE = 1 $\|C\backslash\{s\}\| = 7\ c_{ij}$ | K1 | 2035 | 77.9 | 0 | 70.25 |
| | K2 | 1949 | 79.92 | 0 | 94.15 |
| | K3 | 2035 | 170.13 | 0 | 88.21 |
| | K4 | 1949 | 180.57 | 0 | 60.27 |
| | K5 | 1949 | 173 | 0 | 60.57 |
| SOE = 2 $\|C\backslash\{s\}\| = 7\ c_{ij}$ | K6 | 2123 | 381.84 | 0 | 476.06 |
| | K7 | 2135 | 336.59 | 0 | 314.06 |
| | K8 | 2258 | 340.94 | 0 | 332.62 |
| | K9 | 2197 | 423.35 | 0 | 741.95 |
| | K10 | 2123 | 384.81 | 0 | 430.96 |
| SOE = 3 $\|C\backslash\{s\}\| = 7\ c_{ij}$ | K11 | 2698 | 490.67 | 0 | 1460.1 |
| | K12 | 2862 | 503.7 | 0 | 311.29 |
| | K13 | 3225 | 721.46 | 0 | 1246.04 |
| | K14 | 2169 | 654.75 | 0 | 379.09 |
| | K15 | 2128 | 709.89 | 0 | 245.1 |
| SOE = 4 $\|C\backslash\{s\}\| = 7\ c_{ij}$ | K16 | 5376 | 1668.84 | 1 | 5874.1 |
| | K17 | 4684 | 1536.69 | 0 | 4067.4 |
| | K18 | 5273 | 2187.08 | 1 | 7200 (gap 90 %) |
| | K19 | 4909 | 1334.81 | 1 | 7200 (gap 91 %) |
| | K20 | 4301 | 1298.63 | 1 | 2286.1 |

**Table 16** PDDR model test results with Kartal instances (lower cleaning times)

| Instances features | Instance # | Optimal value | CPU time (Gurobi) | # of cleaned arcs |
|---|---|---|---|---|
| SOE $= 1$ $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K1′ | 2035 | 95.83 | 0 |
| | K2′ | 1949 | 124.7 | 0 |
| | K3′ | 2035 | 102.15 | 0 |
| | K4′ | 1949 | 99.36 | 0 |
| | K5′ | 1949 | 137.65 | 0 |
| SOE $= 2$ $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K6′ | 2123 | 387.9 | 0 |
| | K7′ | 2135 | 426.63 | 0 |
| | K8′ | 2258 | 431.03 | 0 |
| | K9′ | 2197 | 358.51 | 0 |
| | K10′ | 2123 | 422.57 | 0 |
| SOE $= 3$ $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K11′ | 2564 | 722.86 | 1 |
| | K12′ | 2860 | 806.2 | 1 |
| | K13′ | 3225 | 624.41 | 1 |
| | K14′ | 2169 | 605.93 | 0 |
| | K15′ | 2128 | 513.3 | 0 |
| SOE $= 4$ $\lvert C\backslash\{s\}\rvert = 7$ $c'_{ij}$ | K16′ | 4775 | 1741.48 | 2 |
| | K17′ | 4320 | 1495.75 | 2 |
| | K18′ | 4759 | 1956.3 | 3 |
| | K19′ | 4321 | 1717.03 | 1 |
| | K20′ | 3502 | 1615.06 | 2 |

**Table 17** Total route times of optimal solutions obtained from the MIPs using Kartal instances with higher cleaning times

| Ins# | Optimal route time by DRR | Optimal route time by PDRR | Ins# | Optimal route time by DRR | Optimal route time by PDRR |
|---|---|---|---|---|---|
| K1 | 44 | 44 | K11 | 53 | 53 |
| K2 | 43 | 44 | K12 | 63 | 75 |
| K3 | 44 | 47 | K13 | 68 | 82 |
| K4 | 43 | 44 | K14 | 46 | 46 |
| K5 | 43 | 44 | K15 | 47 | 52 |
| K6 | 48 | 48 | K16 | 109 | 117 |
| K7 | 50 | 63 | K17 | 82 | 85 |
| K8 | 51 | 52 | K18 | 110 | 110 |
| K9 | 49 | 49 | K19 | 90 | 93 |
| K10 | 48 | 48 | K20 | 101 | 115 |

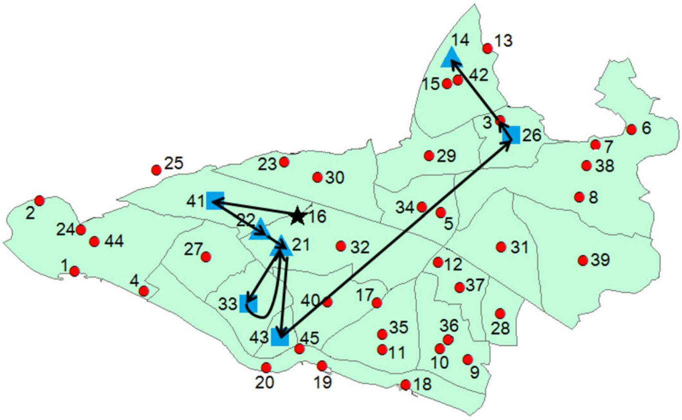


**Fig. 9** Optimal route of instance K7 for DRR

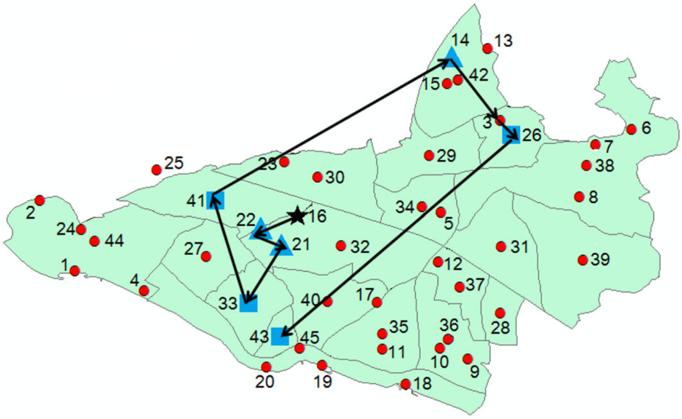


**Fig. 10** Optimal route of instance K7 for PDRR

Out of 20 instances of Kartal with high cleaning effort, optimal routes found by the first and second model are the same for seven of them. Instance K7 is one of them with different optimal routes and its solutions are presented in Figs. 9 and 10. Node 14 is relatively far from other critical nodes and from the supply node, and it is usually the last visited node for Kartal instance. However in the weighted version of the problem, node 14 is visited in the fifth order instead of seventh for instance K7. Hence, the weights, together with the other features of the network, can change the optimal route.

We test our second heuristic, Prioritized Routing with Shortest Paths, with Kartal data set. PRSP solves all instances except one to optimality in around 5 s. The only nonoptimal solution, instance K20′, has a gap of 3.94 % as seen in Table 18.

**Table 18** Performance of PRSP heuristic on Kartal instances with higher and lower cleaning times

| Ins# | Gap % | CPU (s) | Ins# | Gap % | CPU (s) |
|---|---|---|---|---|---|
| K1 | 0.00 | 5.23 | K1′ | 0.00 | 5.17 |
| K2 | 0.00 | 4.86 | K2′ | 0.00 | 5.04 |
| K3 | 0.00 | 5.08 | K3′ | 0.00 | 5.19 |
| K4 | 0.00 | 5.23 | K4′ | 0.00 | 5.19 |
| K5 | 0.00 | 5.03 | K5′ | 0.00 | 5.09 |
| K6 | 0.00 | 5.17 | K6′ | 0.00 | 4.87 |
| K7 | 0.00 | 4.86 | K7′ | 0.00 | 4.95 |
| K8 | 0.00 | 4.65 | K8′ | 0.00 | 4.55 |
| K9 | 0.00 | 4.52 | K9′ | 0.00 | 4.58 |
| K10 | 0.00 | 4.77 | K10′ | 0.00 | 4.79 |
| K11 | 0.00 | 4.82 | K11′ | 0.00 | 4.83 |
| K12 | 0.00 | 4.74 | K12′ | 0.00 | 4.77 |
| K13 | 0.00 | 4.64 | K13′ | 0.00 | 5.08 |
| K14 | 0.00 | 4.38 | K14′ | 0.00 | 4.45 |
| K15 | 0.00 | 4.64 | K15′ | 0.00 | 4.50 |
| K16 | 0.00 | 5.08 | K16′ | 0.00 | 4.84 |
| K17 | 0.00 | 5.04 | K17′ | 0.00 | 5.15 |
| K18 | 0.00 | 4.93 | K18′ | 0.00 | 4.59 |
| K19 | 0.00 | 4.84 | K19′ | 0.00 | 4.50 |
| K20 | 0.00 | 4.71 | K20′ | 3.94 | 4.50 |

### *5.3.2 Bakırköy instances*

The CPU time difference between DRR and PDRR model, which we already observe on Kartal set, is also seen with Bakırköy instances. Especially when the earthquake severity levels are high, it gets harder to reach optimality in a reasonable amount of time.

As seen in Table 19, optimal solutions are found for 9 out of 20 instances when the cleaning effort is high and from Table 20 we see that half of instances are solved to optimality when the cleaning effort is low. As stated in the analyses of the first model, the locations of schools have a tremendous effect on the solutions times.

When the critical set consist of hospitals, we reach the optimal solutions 2 h as seen in Tables 21 and 22. To evaluate the effect of the weights to the optimal route, we compare the total route times obtained by the DRR and PDRR models on Bakırköy instances where critical nodes are only hospitals and the cleaning effort is high. We see that 5 out of 20 instances have the same total route time for DRR and PDRR models.

With critical node set of hospitals, we have the optimal solutions of Bakırköy instance thus we are able to evaluate the solutions obtained by PRSP heuristic. The instances are solved in 12 s on the average and all optimal solutions are found except for instance B17′ whose gap is 0.6 %. When schools are taken as the critical nodes, 19

**Table 19** PDDR model test results with Bakırköy instances (schools, higher cleaning times)

| Instance features | Instance # | Best objective | Gap % | CPU (s) Gurobi | # of cleaned arcs |
|---|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c_{ij}$ | B1 | 2779 | 0.00 | 1721.08 | 0 |
| | B2 | 3439 | 0.00 | 3738.91 | 1 |
| | B3 | 2779 | 0.00 | 2442.78 | 0 |
| | B4 | 2847 | 0.00 | 3051.61 | 0 |
| | B5 | 2779 | 0.00 | 6052.22 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c_{ij}$ | B6 | 2779 | 0.00 | 3605.48 | 0 |
| | B7 | 3591 | 77.28 | 7200 | 0 |
| | B8 | 2779 | 0.00 | 4499.47 | 0 |
| | B9 | 3556 | 0.00 | 5646.93 | 0 |
| | B10 | 2779 | 0.00 | 3676.88 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c_{ij}$ | B11 | 3802 | 90.90 | 7200 | 0 |
| | B12 | 4376 | 89.90 | 7200 | 0 |
| | B13 | 4131 | 91.43 | 7200 | 0 |
| | B14 | 4017 | 89.15 | 7200 | 0 |
| | B15 | 4199 | 92.99 | 7200 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 8$ (schools) $c_{ij}$ | B16 | 5125 | 95.82 | 7200 | 0 |
| | B17 | 3928 | 97.66 | 7200 | 0 |
| | B18 | 6105 | 98.33 | 7200 | 0 |
| | B19 | 4426 | 100.00 | 7200 | 0 |
| | B20 | 4030 | 96.87 | 7200 | 0 |

**Table 20** PDDR model test results with Bakırköy instances (schools, lower cleaning times)

| Instance features | Instance # | Best objective | Gap% | CPU time Gurobi | # of cleaned arcs |
|---|---|---|---|---|---|
| SOE = 1 $\lvert C\setminus\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B1′ | 2779 | 0.00 | 1629.69 | 0 |
| | B2′ | 3136 | 0.00 | 2778.12 | 2 |
| | B3′ | 2779 | 0.00 | 2711.83 | 0 |
| | B4′ | 2847 | 0.00 | 1793.75 | 0 |
| | B5′ | 2779 | 0.00 | 2862.37 | 0 |
| SOE = 2 $\lvert C\setminus\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B6′ | 2779 | 0.00 | 1946.02 | 0 |
| | B7′ | 3183 | 0.00 | 3475.90 | 1 |
| | B8′ | 2779 | 0.00 | 3490.22 | 0 |
| | B9′ | 3183 | 0.00 | 3325.31 | 1 |
| | B10′ | 2779 | 0.00 | 3347.79 | 0 |
| SOE = 3 $\lvert C\setminus\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B11′ | 3802 | 83.90 | 7200 | 0 |
| | B12′ | 4232 | 86.51 | 7200 | 2 |
| | B13′ | 4045 | 88.23 | 7200 | 1 |
| | B14′ | 3898 | 81.35 | 7200 | 1 |
| | B15′ | 4028 | 84.51 | 7200 | 2 |
| SOE = 4 $\lvert C\setminus\{s\}\rvert = 8$ (schools) $c'_{ij}$ | B16′ | 4927 | 97.16 | 7200 | 1 |
| | B17′ | 3725 | 99.19 | 7200 | 1 |
| | B18′ | 5303 | 98.08 | 7200 | 3 |
| | B19′ | 4452 | 97.87 | 7200 | 3 |
| | B20′ | 4194 | 96.76 | 7200 | 2 |

**Table 21** PDDR model test results with Bakırköy instances (hospitals, higher cleaning times)

| Instance features | Instance # | Optimal value | CPU time Gurobi | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B1 | 1979 | 338 | 0 |
| | B2 | 1876 | 355.64 | 0 |
| | B3 | 1884 | 531.34 | 0 |
| | B4 | 1862 | 534.37 | 0 |
| | B5 | 1920 | 318.56 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B6 | 1884 | 403.22 | 0 |
| | B7 | 1907 | 541.26 | 0 |
| | B8 | 1924 | 642.64 | 0 |
| | B9 | 1897 | 722.25 | 0 |
| | B10 | 1876 | 704.38 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B11 | 1875 | 2241.83 | 0 |
| | B12 | 1884 | 1663.46 | 0 |
| | B13 | 2169 | 1925.84 | 0 |
| | B14 | 2362 | 2419.83 | 0 |
| | B15 | 1970 | 2080.29 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c_{ij}$ | B16 | 2430 | 4135.24 | 0 |
| | B17 | 2671 | 4890.86 | 0 |
| | B18 | 2361 | 4833.86 | 0 |
| | B19 | 2672 | 6275.62 | 0 |
| | B20 | 2483 | 4944.91 | 0 |

instances are solved to optimality by the PDRR model. PRSP heuristic solves 18 of them to optimality and the gap for instance B12′ is 0.12 %. For the rest, 21 instances, that cannot be solved to optimality due to the time limit, the PRSP heuristic finds the same or better solutions compared to the PDRR model, in 18.3 s on the average. These results are summarized in Table 23. Positive gap indicates that the PRSP heuristic performs better.

## 6 Conclusion

In this study, we proposed solution methodologies for Prioritized Debris Removal Problem in the Response Phase. The MIP formulation suggested for the DRR problem is proved to be more efficient than the one developed by Sahin et al. (2015) by the comparison of Kartal solutions. The computational studies with a larger network has shown that the number and locations of the critical nodes have tremendous effect on

**Table 22** PDDR model test results with Bakırköy instances (hospitals, lower cleaning times)

| Instance features | Instance # | Optimal value | CPU time Gurobi | # of cleaned arcs |
|---|---|---|---|---|
| SOE = 1 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B1′ | 1979 | 316.54 | 0 |
| | B2′ | 1876 | 294.52 | 0 |
| | B3′ | 1884 | 335.47 | 0 |
| | B4′ | 1862 | 255.17 | 0 |
| | B5′ | 1920 | 287.19 | 0 |
| SOE = 2 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B6′ | 1884 | 348.97 | 0 |
| | B7′ | 1907 | 444.67 | 0 |
| | B8′ | 1924 | 410.58 | 0 |
| | B9′ | 1897 | 438.99 | 0 |
| | B10′ | 1876 | 371.99 | 0 |
| SOE = 3 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B11′ | 1875 | 1394.25 | 0 |
| | B12′ | 1884 | 1330.03 | 0 |
| | B13′ | 2169 | 1254.56 | 0 |
| | B14′ | 2362 | 1145.38 | 0 |
| | B15′ | 1970 | 1210.83 | 0 |
| SOE = 4 $\lvert C\backslash\{s\}\rvert = 7$ (hospitals) $c'_{ij}$ | B16′ | 2426 | 4963.48 | 1 |
| | B17′ | 2655 | 4981.34 | 1 |
| | B18′ | 2361 | 4518.54 | 0 |
| | B19′ | 2570 | 5621.99 | 1 |
| | B20′ | 2483 | 6060.65 | 0 |

the solution times. To obtain near optimal solutions in a timely manner we suggested an efficient heuristic which basically solves a modified TSP among the critical nodes.

Our main contribution to the literature is the second problem and its corresponding model. To the best of the authors' knowledge, there is no study considering node priorities in this problem. The MIP model formulated for this problem resulted in higher CPU times due the necessity of knowing exact times of the debris removal and arrivals to the critical nodes. To obtain solutions of good quality we developed an efficient heuristic.

In this study, we focus on earthquakes and consider the routing problem for small provinces where each municipality has only one vehicle which can clean the blocked arcs. For other types of disasters that impact a larger area, the single vehicle assumption would not be valid, therefore routing with heterogeneous fleet can be a realistic extension. Moreover, the critical points which constitute a subset of nodes in our study, can be taken as a subset of edges and nodes.

**Table 23** Comparison of PDRR model and PRSP heuristic performances with Bakırköy instances

| Critical nodes | Cleaning time | Instance # | PDRR model solution status | PRSP heuristic | |
|---|---|---|---|---|---|
| | | | | Opt/gap | Avg CPU time |
| Hospitals | High | B1–B20 | Opt | Opt | 11.74 |
| | Low | B1′–B16′, B18′–B′20 | Opt | Opt | 11.98 |
| | | B17′ | Opt | −0.60 | 10.68 |
| Schools | High | B1–B6, B8–B10 | Opt | Opt | 18.32 |
| | | B7 | Nonopt | 0.00 | 19.48 |
| | | B11 | Nonopt | 0.00 | 17.54 |
| | | B12 | Nonopt | 0.00 | 17.99 |
| | | B13 | Nonopt | 0.15 | 19.25 |
| | | B14 | Nonopt | 0.00 | 18.15 |
| | | B15 | Nonopt | 0.00 | 19.07 |
| | | B16 | Nonopt | 1.68 | 17.20 |
| | | B17 | Nonopt | 5.17 | 13.59 |
| | | B18 | Nonopt | 12.12 | 25.79 |
| | | B19 | Nonopt | 0.56 | 19.39 |
| | | B20 | Nonopt | 0.00 | 14.41 |
| | Low | B1′–B10′ | Opt | Opt | 17.99 |
| | | B11′ | Nonopt | 0.00 | 18.08 |
| | | B12′ | Nonopt | −0.12 | 18.37 |
| | | B13′ | Nonopt | 0.00 | 19.41 |
| | | B14′ | Nonopt | 0.00 | 19.68 |
| | | B15′ | Nonopt | 0.00 | 19.37 |
| | | B16′ | Nonopt | 2.09 | 16.70 |
| | | B17′ | Nonopt | 0.00 | 17.54 |
| | | B18′ | Nonopt | 0.19 | 18.33 |
| | | B19′ | Nonopt | 4.69 | 18.29 |
| | | B20′ | Nonopt | 7.94 | 16.90 |

Gap = (PDRR solution − PRSP solution)/PDRR solution × 100

**Acknowledgments** We sincerely thank the two anonymous referees for contributing to the improvement of our paper. The authors gratefully acknowledge the support provided to this work by the Turkish Academy of Sciences and TUBITAK through project 112M225.

## References

Afshar A, Haghani A (2012) Modeling integrated supply chain logistics in real-time large-scale disaster relief operations. Soc Econ Plan Sci 46(4):327–338

Aksu DT, Ozdamar L (2014) A mathematical model for post-disaster road restoration: enabling accessibility and evacuation. Transp Res Part E Logist Transp Rev 61:56–67

Altay N, Green WG (2006) Or/ms research in disaster operations management. Eur J Oper Res 175(1):475–493

Apte A (2010) Humanitarian logistics: a new field of research and action, vol 7. Now Publishers, Boston

Averbakh I (2012) Emergency path restoration problems. Discrete Optim 9(1):58–64

Averbakh I, Pereira J (2012) The flowtime network construction problem. Iie Trans 44(8):681–694

Averbakh I, Pereira J (2015) Network construction problems with due dates. Eur J Oper Res 244(3):715–729

Balçık B, Beamon BM (2008) Facility location in humanitarian relief. Int J Logist 11(2):101–121

Barbarosoglu G, Arda Y (2004) A two-stage stochastic programming framework for transportation planning in disaster response. J Oper Res Soc 55(1):43–53

Berkoune D, Renaud J, Rekik M, Ruiz A (2012) Transportation in disaster response operations. Soc Econ Plan Sci 46(1):23–32

Campbell AM, Vandenbussche D, Hermann W (2008) Routing for relief efforts. Transp Sci 42(2):127–145

Caunhye AM, Nie X, Pokharel S (2012) Optimization models in emergency logistics: a literature review. Soc Econ Plan Sci 46(1):4–13

Çelik M, Ergun Ö, Keskinocak P (2015) The post-disaster debris clearance problem under incomplete information. Oper Res 63(1):65–85

Chang MS, Tseng YL, Chen JW (2007) A scenario planning approach for the flood emergency logistics preparation problem under uncertainty. Transp Res Part E Logist Transp Rev 43(6):737–754

Chen YW, Tzeng GH (1999) A fuzzy multi-objective model for reconstructing the post-quake road-network by genetic algorithm. Int J Fuzzy Syst 1(2):85–95

de la Torre LE, Dolinskaya IS, Smilowitz KR (2012) Disaster relief routing: integrating research and practice. Soc Econ Plan Sci 46(1):88–97

Duran S, Gutierrez MA, Keskinocak P (2011) Pre-positioning of emergency items for care international. Interfaces 41(3):223–237

EM-DAT UoLB CRED (2013) Disasters in numbers. http://www.cred.be/sites/default/files/Disasters-in-numbers-2013.pdf. Accessed June 2014

FEMA (2010) Public assistance debris monitoring guide. http://www.fema.gov/pdf/government/grant/pa/fema327debrismonitoring.pdf. Accessed June 2014

Feng CM, Wang TC (2003) Highway emergency rehabilitation scheduling in post-earthquake 72 hours. J 5th East Asia Soc Transp Stud 5:3276–3285

Fetter G, Rakes T (2012) Incorporating recycling into post-disaster debris disposal. Soc Econ Plan Sci 46(1):14–22

Galindo G, Batta R (2013) Review of recent developments in OR/MS research in disaster operations management. Eur J Oper Res 230(2):201–211

Haghani A, Oh SC (1996) Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. Transp Res Part A Policy Pract 30(3):231–250

Hu ZH, Sheu JB (2013) Post-disaster debris reverse logistics management under psychological cost minimization. Transp Res Part B Methodol 55:118–141

Huang M, Smilowitz K, Balcik B (2012) Models for relief routing: equity, efficiency and efficacy. Transp Res Part E Logist Transp Rev 48(1):2–18

Kılcı F, Kara BY, Bozkaya B (2015) Locating temporary shelter areas after an earthquake: a case for Turkey. Eur J Oper Res 243(1):323–332

Lin YH, Batta R, Rogerson PA, Blatt A, Flanigan M (2011) A logistics model for emergency supply of critical items in the aftermath of a disaster. Soc Econ Plan Sci 45(4):132–145

Lush T (2010) 98 % of haiti quake debris remains. http://www.chinapost.com.tw/international/americas/2010/09/14/272484/98-of.htm. Accessed April 2014

Mete HO, Zabinsky ZB (2010) Stochastic optimization of medical supply location and distribution in disaster management. Int J Prod Econ 126(1):76–84

Murphy D (2010) Haiti earthquake: Small Port-au-Prince airport strained by aid demand. http://www.csmonitor.com/World/Global-News/2010/0115/Haiti-earthquake-Small-Port-au-Prince-airport-strained-by-aid-demand. Accessed February 2014

Nurre SG, Sharkey TC (2014) Integrated network design and scheduling problems with parallel identical machines: complexity results and dispatching rules. Networks 63(4):306–326

Nurre SG, Cavdaroglu B, Mitchell JE, Sharkey TC, Wallace WA (2012) Restoring infrastructure systems: an integrated network design and scheduling (INDS) problem. Eur J Oper Res 223(3):794–806

Özdamar L, Ertem MA (2014) Models, solutions and enabling technologies in humanitarian logistics. Eur J Oper Res

Pramudita A, Taniguchi E, Qureshi AG (2014) Location and routing problems of debris collection operation after disasters with realistic case study. Proc Soc Behav Sci 125:445–458

Rawls CG, Turnquist MA (2009) Pre-position of emergency supplies for disaster response. Transp Res Part E

Sahin H, Kara BY, Karasan OE (2015) Debris removal during disaster response: a case for turkey. Soc Econ Plan Sci

Samii R, Van Wassenhove LN, Kumar K, Becerra-Fernandez I (2002) Choreographer of disaster management: preparing for tomorrows disasters. INSEAD, Fontainebleau, No. 06 /2002-5039

Sharif MT, Salari M (2015) A grasp algorithm for a humanitarian relief transportation problem. Eng Appl Artif Intell 41:259–269

Sheu JB (2007) Challenges of emergency logistics management. Transp Res Part E Logist Transp Rev 43(6):655–659

Stephenson J (2008) Hurricane Katrina: continuing debris removal and disposal issues. Diane, London

Tübitak (2005) Türkiye ulusal deprem araştırmaları programı. http://www.tubitak.gov.tr/tubitakcontentfiles/ARDEB. Accessed June 2014

Tzeng GH, Cheng HJ, Huang TD (2007) Multi-objective optimal planning for designing relief delivery systems. Transp Res Part E Logist Transp Rev 43(6):673–686

Van Hentenryck P, Bent R, Coffrin C (2010) Strategic planning for disaster recovery with stochastic last mile distribution. In: Integration of AI and OR techniques in constraint programming for combinatorial optimization problems. Springer, New York, pp 318–333

Wang H, Du L, Ma S (2014) Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake. Transp Res Part E Logist Transp Rev 69:160–179

Yi W, Kumar A (2007) Ant colony optimization for disaster relief operations. Transp Res Part E Logist Transp Rev 43(6):660–672

Yi W, Özdamar L (2007) A dynamic logistics coordination model for evacuation and support in disaster response activities. Eur J Oper Res 179(3):1177–1193