A decomposition-based heuristic for a multicrew coordinated road restoration problem[1]

Vahid Akbari[a,*], Mir Ehsan Hesam Sadati[b,c], Ramez Kian[d]

[a]Nottingham University Business School, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom
[b]Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
[c]Smart Mobility and Logistics Lab, Sabanci University, Istanbul, Turkey.
[d]Nottingham Business School, Nottingham Trent University, Nottingham NG1 4FQ, United Kingdom

Abstract

Natural disasters disrupt the connectivity of road networks by blocking road segments, which impedes efficient distribution of relief materials to the affected area. We study the problem of finding coordinated paths for clearing teams so that the connectivity of the road network is regained in the shortest time. We provide an efficient novel heuristic algorithm for this problem. In our algorithm, the problem is first pre-processed to define a binary problem to generate initial solutions, and then several rich and problem-specific neighborhood search moves are applied to improve the derived initial solutions. We provide several analytical results which facilitate the design of our algorithm. The performance of our proposed algorithm is assessed by different numerical experiments, and a comparison with existing algorithms from the literature using instances from Istanbul road networks. The results demonstrate that our algorithm performs notably better, both in terms of speed, and proximity to optimal solution.

Keywords: Disaster management, humanitarian logistics, road restoration, network connectivity, heuristics, relief distribution

1. Introduction

Over the last thirty years, there has been an exponential growth in the number and magnitude of disasters, affecting approximately on average 300 million people per annum and costing close to $157 billion worth of economic damage per annum since the 1990s (Özdamar & Ertem 2015). In the wake of this growth, major Transportation organizations of the USA such as Departments of Transportation (DoTs) and the Transportation Research board promoted research which provides further insight in supporting disaster

[1]cite this study as: Akbari, V., Sadati, M. E. H., and Kian, R. (2021). A decomposition-based heuristic for a multicrew coordinated road restoration problem.Transportation Research Part D: Transport and Environment, 95:102854.

*Corresponding author

Email addresses: vahid.akbari@nottingham.ac.uk (Vahid Akbari), msadati@sabanciuniv.edu (Mir Ehsan Hesam Sadati), ramez.kian@ntu.ac.uk (Ramez Kian)

response operations. (Kim *et al.* 2019). While the DoTs are responsible for a large portfolio of transportation decisions under normal conditions, they are also responsible for providing a response to unexpected natural or man-made events that disrupts the transportation network (Renne *et al.* 2020). Mitigating negative effects of such disasters through disaster management is therefore an important area of research. Sudden-onset of natural disasters, such as earthquakes, might not be preventable, but in order to minimize the post-disaster risks of having a high number of casualties, appropriate steps should be taken in response to the incident. For instance, to minimize socio-economic losses, such as loss of human lives and valuables, a sufficient amount of relief material or supplies must be distributed effectively and efficiently immediately after a catastrophe.

One of the critical problems that occur after such events is the road network disruptions. A set of road segments is generally expected to be significantly damaged and hence, unusable during the immediate response phase. This blockage of the roads can divide the network into several isolated parts; preventing delivery of relief materials to the affected area, and from the affected area to crucial locations such as shelters and hospitals. For instance, Golla *et al.* (2020) investigated accessibility of urban locations in a post-disaster condition where collapsed buildings can obstruct roads and prevent accessibility to some certain locations. The connectivity of the road network plays a key role in disaster management to avoid making cities shut-down. The importance of road network connectivity after a disaster has been addressed and studied in Zhou *et al.* (2019). Expediting the road recovery procedure of a damaged road network in a post-disaster condition can improve the resilience of the infrastructure networks (Sun & Zhang 2020) as well. In the response phase immediately after occurrence of a disaster, for effective and efficient distribution of relief materials such as blankets, hygiene kits, water, food, clothes and medicines, a number of road clearance teams are dispatched to affected regions. They are responsible for identifying and recovering a sub-set of the blocked roads with the purpose of reconnecting the entire road network.

Examples of post-disaster road damages and inaccessibility are abundant, and several disaster scenarios are studied as case studies in the literature of road restoration. Tuzun Aksu & Ozdamar (2014) proposed a model to address the accessibility maximisation problem through road restoration for survivor evacuation and tested their model on two districts of Istanbul. Caunhye *et al.* (2020) implemented a robust route restoration model and tested their model on the data sets from 2015 Gorkha earthquake in Nepal. Coco *et al.* (2020) studied the problem of restoring a set of disruptions in an urban road network while ensuring the connectivity of the network and tested their bi-objective model on realistic instances built from the road network map of Troyes city in France. Barbalho *et al.* (2021) proposed a Greedy Randomized Adaptive Search procedure and an Iterated Local Search algorithm to address the multi-period Work-troops Scheduling Problem (WSP) and tested their algorithms on instances from Port-au-Prince city in Haïti after the 2010 earthquake. Cartes *et al.* (2021) applied an approach to recover the resilience of a road after a disaster to a road located in the south of Chile.

The information regarding damaged roads and the amount of accumulated debris on such roads can be gathered using Volunteered Geographic Information (VGI) or satellite images. The information collection phase of the immediate post-disaster stage has been addressed in some studies (see Oruc & Kara (2018) and Sokat *et al.* (2018)). In our study however, we assume that all the blockage information is collected and the exact time needed for clearing teams to open the blocked edges is known. Given that the blockage information on the blocked links is available, we furthermore assume that the teams are homogeneous meaning that the required time to open a blocked edge is the same for all of the teams. These teams are equipped with the necessary tools to either rapidly clear or repair the blocked edges; and if that is not possible, bypass them by alternative modes such as temporary roads. From this point forward, we also refer to these equipped teams by *vehicles* as well.

With a connectivity objective, we study the post-disaster debris clearance problem in which the goal is to identify a subset of the obstructed roads that should be unblocked in order to regain the connectivity of the network. Furthermore, the solution should construct coordinated paths for the vehicles to open these roads in the shortest time i.e, minimizing the makespan. Note that since a blocked edge is non-traversable unless it is entirely restored or unblocked, the paths should be coordinated to prevent traversing blocked edges without opening them. Moreover, a blocked road can be opened by a team and traversed by others. As is expected, while an edge is undergoing the unblocking procedure by a vehicle, other vehicles cannot enter it. As a result, the arrival time of the vehicles to the blocked edges, as well as the time in which the unblocking procedure of the blocked edges is finished should be calculated to prevent traversal time conflicts. However, since a node can be visited by the same vehicle multiple times and also the same edges can be traversed by multiple vehicles, time calculations are very complicated for this problem. Moreover, since some of the blocked edges are opened during the process, shortest path distances between each pair of the nodes might change which can further complicate the timing calculations.

Kasaei & Salman (2016) studied the single-crew case of this problem. They proposed a mathematical model as well as a heuristic algorithm to solve this problem. However, the single-crew case of this problem does not address timing conflicts as it does not happen when only one vehicle is considered. As a result, in this study, the above-mentioned issue is not considered. In Akbari & Salman (2017b), a Mixed Integer Programming (MIP) model to solve the multicrew version of the described problem is developed. However, their model could not find feasible solutions to even moderate-sized instances as the timing constraints restrain its performance. Hence, a relaxation of the formulation is developed to generate initial solutions for a local search algorithm process. Then, the solution of the relaxed problem is changed to a feasible solution and improved by local search procedures to obtain a feasible solution for the original problem. Their problem is called the Multi-vehicle Synchronized Arc Routing for Connectivity Problem (K-ARCP) which aims to provide practical solutions for minimizing time for connecting the road network.

In this article, we contribute to the literature by deeply analysing the optimal features of the K-ARCP through a number of propositions and procedures. Then we focus on more complex instances of this problem compared to those of the literature and develop a MIP-based heuristic with Rich Local Search (MIP-RLS) moves which performs considerably better than the proposed local search algorithm in Akbari & Salman (2017b). In order to generate initial solutions for the MIP-RLS, we pre-process the problem to form a binary programming problem. Using this problem along with a feasibility step, we generate the initial solutions. We show that the generated solutions can be optimal in some of the instances. Several new and rich problem-specific neighborhood search moves are developed to improve the initial solutions. The performance of the MIP-RLS is tested and compared with the proposed algorithm in Akbari & Salman (2017b) on Istanbul road network data. The results show that the MIP-RLS is capable of solving larger instances in a significantly shorter time. Even in the instances where the method from the literature is unable to find feasible solutions in a given time-limit, on average, the MIP-RLS finds good solutions in less than half of that time-limit. Our proposed algorithm can thus be used in the response phase of the post-disaster decision support systems.

The remainder of the paper is organized as follows. In Section 2, we provide a literature review of the related studies that address the utilization of humanitarian logistics in disaster management response, followed by different algorithms used for post-disaster response operations. Section 3 provides the problem definition and in Section 4, we present a short overview of the proposed approach in the literature and then state our solution approach. The process of the data generation and the results are given in Section 5. Finally, we close by some concluding remarks in Section 6.

2. Literature review

Within the context of disaster management, disasters are categorized into four main phases including mitigation, preparation, response and recovery (Ahmadi *et al.* 2015). Mitigation and preparation are part of the pre-disaster phase whereas response and recovery happens in the post-disaster stage. In the pre-disaster phase, planning for risk mitigation and prevention helps in minimizing damage in areas affected by disaster. Pre-disaster preparedness time for disasters such as floods and storms is more than that of earthquakes as the predictability related to the latter is uncertain. Therefore, inventory management and equipment pre-positioning can enable fast and effective relief distribution from pre-stocked relief materials such as food, water, clothes and medicines to populations affected from floods and storms (Mete & Zabinsky 2010, Duran *et al.* 2011, Özdamar & Ertem 2015). However, it is mainly the logistics performance during the post-disaster phase that determines the socio-economic impact occurring as a result of earthquake. To minimize this socio-economic impact, post-disaster phases should involve effective and efficient distribution of relief material distribution to the affected area, debris collection, casualty transportation to hospitals and mass evacuation to shelters. Such activities enhance post-disaster survival rates. However, successful implementation of them requires good logistical support.

There have been many review articles published in the last decade with a focus on humanitarian logistics in post-disaster management. Anaya-Arenas *et al.* (2014) reviewed articles on disaster management and based on their findings, they claimed that emergency repair of the damages on a road network is the least popular subject. Çelik (2016) provided a review of the articles that addressed network restoration and recovery in humanitarian operations. Minas *et al.* (2020) published a review article to study the application of operations research models that cover all of these variations of emergency response problems. The K-ARCP falls into different categorizations; such as traditional arc routing problems or problems with a focus on road restoration and upgrading in both disastrous and non-disastrous situations. In general, the K-ARCP can be seen as an application of heuristics in routing problems as well. In the following paragraphs, we first discuss the problems with a focus on road restoration and recovery and then give the preliminary studies on the K-ARCP.

In the studies that focus on upgrading the road network or improvement of the accessibility after an earthquake, there are two basic approaches. The first approach concentrates on the selection of road segments that should be upgraded or repaired and ignores the routing of the vehicles. For instance Duque & Sörensen (2011) studied a post-disaster problem with a budget constraint. In their problem, with a goal of minimizing the weighted sum of the required time to travel from rural areas to the nearest emergency centre, some of the blocked roads are identified to be recovered. They have used a greedy randomized adaptive search procedure (GRASP) to generate initial solutions and improved their obtained solutions using a variable neighborhood search (VNS) algorithm. In another study, Cavdaroglu *et al.* (2013) developed a mathematical formulation as well as a heuristic solution method for a problem in which integration of the restoration planning and scheduling decisions in order to restore essential services provided by interdependent infrastructure systems is considered.

The other group of related studies under the road restoration and recovery domain are those that focus on both road network restoration and distribution of relief items and routing of the team(s). Xu & Song (2015) built a model for joint scheduling of post-disaster road restoration and relief distribution to optimize the efficiency of life-saving commodity delivery. Their model is formulated as a mixed-integer multiple commodity network flow programming problem and an ant colony optimization (ACO) approach is used to tackle the problem. Duque *et al.* (2016) developed a Dynamic Programming (DP) approach, which is able to provide the schedule and route of a repair crew for small scale instances. The authors also developed an iterated greedy-randomized constructive procedure to solve large-scale instances. The objective of this study is to minimize the summation of the time in which demand nodes are accessible from the depot. Moreno *et al.* (2019) built upon the problem in Duque *et al.* (2016) and introduced The Crew Scheduling and Routing Problem (CSRP) that targets minimizing the time that affected areas remain inaccessible. In their study, a Branch-and-Benders-cut (BBC) algorithm is developed to tackle this problem. Moreno *et al.* (2020a) studied the same problem and developed two new metaheuristics to solve the CSRP. Using a new hybrid BBC approach they have found new optimal solutions and improved bounds for benchmark instances of the problem.

Ajam *et al.* (2019) addressed a simultaneous road restoration and relief distribution problem. In this study, both of these actions are operated by one crew. A set of pre-identified nodes referred to as critical locations should receive the relief items with a latency minimization objective function. The latency of a critical node is defined as the time it takes from the beginning of the planning horizon until that node is served. Shin *et al.* (2019) and Briskorn *et al.* (2020) also addressed the integration of road restoration and relief distribution operations. Shin *et al.* (2019) proposed a mixed integer linear programming model to provide a plan for the road clearance and relief distribution teams such that damaged roads are repaired and relief items are supplied to demand locations. Briskorn *et al.* (2020) addressed a problem to determine the blocked edges to unblock and the delivery amount from each supply node to each demand node in each period, while minimizing the duration of the road clearance activities. Other objectives and operational challenges have been incorporated in such studies. For instance, Li & Teo (2019) developed a bi-level mathematical model for post-earthquake network repair problem wherein the repair crew assignment and routing decisions are made in the upper level, while fairness of the relief allocation and finding shortest path to have quick response are the two other objectives for the decision makers of the lower level. Similarly, Li *et al.* (2020) integrated logistics support scheduling with repair crew scheduling and routing in post-disaster network restoration. Our study is different from all of the above in terms of our objective function and existence of multiple crews.

While most of the studies addressing the post-disaster road restoration problems are of a deterministic nature, some of the studies assume that the unblocking time of the blocked edges is stochastic. Çelik *et al.* (2015) developed a multi-period stochastic problem that captures the dynamic post-disaster status in which road clearing times are assumed uncertain, and limited information on the debris amounts along the roads is updated as clearance activities proceed. The solution gives the sequence of the blocked roads that should be unblocked in each of the periods. In their study, they first developed a Markov decision process (MDP) model and then they suggested a search tree and heuristic pruning, which were tested on both random and real data. Sayarshad *et al.* (2020) proposed a dynamic non-myopic debris clearance problem that incorporates re-positioning of equipment items in a post-disaster stochastic environment to make connectivity between supply and demand nodes. Sanci & Daskin (2019) integrated restoration and location models for effective disaster response. They used two-stage stochastic programming to incorporate disaster scenarios, and solved the model by sample average approximation with concentration sets. Sanci & Daskin (2021) developed an integer L-shaped algorithm based on a branch-and-cut procedure to solve the same problem. Our study is different from the ones mentioned in this section as we are considering multiple teams and different objectives.

Some studies incorporated online optimization to address the uncertainties. In an online optimization context, Akbari & Shiri (2021) studied the post-disaster relief distribution problem in which the blocked edges are not known in advance and are only revealed when they are observed by the relief distribution teams. In this study, the relief distribution teams are not able to recover these blocked edges and should find a way to bypass them when they are observed. Shiri *et al.* (2020) studied a post-disaster heterogeneous multi-team search-and-rescue problem modeled on an undirected graph involving non-recoverable online blocked edges. In this study, the search-and-rescue time for each critical node is also online. The blocked edges cannot be recovered here, and only search-and-rescue operations in pre-identified nodes is considered.

The single-crew case of the K-ARCP (ARCP) was studied and introduced in Kasaei & Salman (2016). In this study, the authors provide a constructive heuristic algorithm for the ARCP. They also introduced a variation of the ARCP that instead of regaining the connectivity of the network, collects maximum prizes by reopening components to the depot node in a given time-limit. In their problem, a prize is assigned to opening each of the components. This variation is referred to as the Prize Collecting ARCP denoted by PC-ARCP. Vodák *et al.* (2018) focused on the single vehicle case too and developed a metaheuristic using ACO that reconnects all the isolated components by dispatching a single vehicle. Note that in both of these studies mentioned above, a single vehicle exists and hence, timing conflicts do not occur.

Akbari & Salman (2017a) proposed a matheuristic to solve the multi-vehicle case of the PC-ARCP which was denoted by KPC-ARCP. In a related study, Morshedlou *et al.* (2018) proposed two MIP models to solve a routing problem in which synchronized planning and scheduling of recovery activities for damaged networks is addressed. They have tested their models on infrastructure network instances from Shelby County in

Tennessee, USA. Moreno *et al.* (2020b) developed three mathematical models for the multicrew scheduling and routing problem using heterogeneous teams and validated their models though instances generated from floods and landslides in Rio de Janeiro. Different from our study, they have incorporated the blockages on the nodes. Finally, Akbari & Salman (2017b) introduced the K-ARCP for the first time. They developed a local search matheuristic procedure to tackle the problem. In our study, we further focus on the complexity of this problem, analyse the optimality features through a number of propositions and develop an efficient approach by developing a MIP-based heuristic with Rich Local Search moves denoted by **MIP-RLS**. We provide through computational experiments to verify that the MIP-RLS significantly outperforms the local search algorithm proposed in Akbari & Salman (2017b). Specifically, for some instances that the local search algorithm is not able to find feasible solutions within a certain time-limit, the MIP-RLS is capable of finding either optimal or very good solutions with a proven negligible optimality gap. We verify our claim by testing and comparing our algorithm with the algorithm presented in Akbari & Salman (2017b) on Istanbul Road Network instances.

3. Problem definition

In order to be consistent with the literature, we define the problem with the same notations given in Akbari & Salman (2017b). We represent the road network of a city or region by an undirected connected graph given as $G = (V, E)$ where V denotes the set of nodes and E gives the set of all the edges prior to an incident. Assuming that the graph is undirected, is a standard assumption in disaster management modelling. This is because in disaster conditions, we can assume that roads can be used in both directions by road restoration teams. For each edge $(i, j) \in E$ there is a time associated with travelling from node i to j denoted by c_{ij}. Given that the road network is modeled by an undirected graph, the traversal times are assumed to be asymmetric. After a disruption occurs in the road network, a subset of the edges denoted by B, $B \subset A$, are blocked and the intact edges together with the nodes set V form the graph $G_B = (V, E \setminus B)$, which is separated into Q components. We assume that the location of these blocked edges is known and is collected from satellite images, open street maps and by the use of helicopters or drones.

Blocked edges are assumed to be non-traversable unless they are fully restored by one of the road clearance teams. The unblocking procedure of edge $(i, j) \in B$ requires an additional b_{ij} units of time to be completed. As a result, traversing and restoring a blocked arc $(i, j) \in B$ for one of the road restoration teams takes $c_{ij} + b_{ij}$ units of time. An important observation is that, after a blocked arc (i, j) is restored by one of the road restoration teams, the other road restoration teams can traverse it within c_{ij} units of time. Since blocked edges are only required to be opened once and their traversal time changes after they are recovered, the problem becomes more complex compared to regular arc or node routing problems. This means that the restoration timing of the blocked links should be considered in the formulation which significantly complicates the problem.

Knowing that G_B consists of Q separated components, let us furthermore denote the q^{th} separated component of G_B by C_q ($q = 1, ..., Q$). We also denote the node and edge sets of the C_q by V_q and E_q, respectively. The goal of the K-ARCP is to find a sub-set of the blocked edges denoted by R, in a way that $G_R = (V, (E \setminus B) \cup R)$ forms a connected graph. The solution to this problem should find the set R and extract a path for each of the road clearance teams starting at the depot. These paths collectively cover R. We also denote the depot node where all the road clearance teams are initially positioned by D. The objective function of the K-ARCP is to minimize the latency or the time in which the last road clearance team finishes their operations.

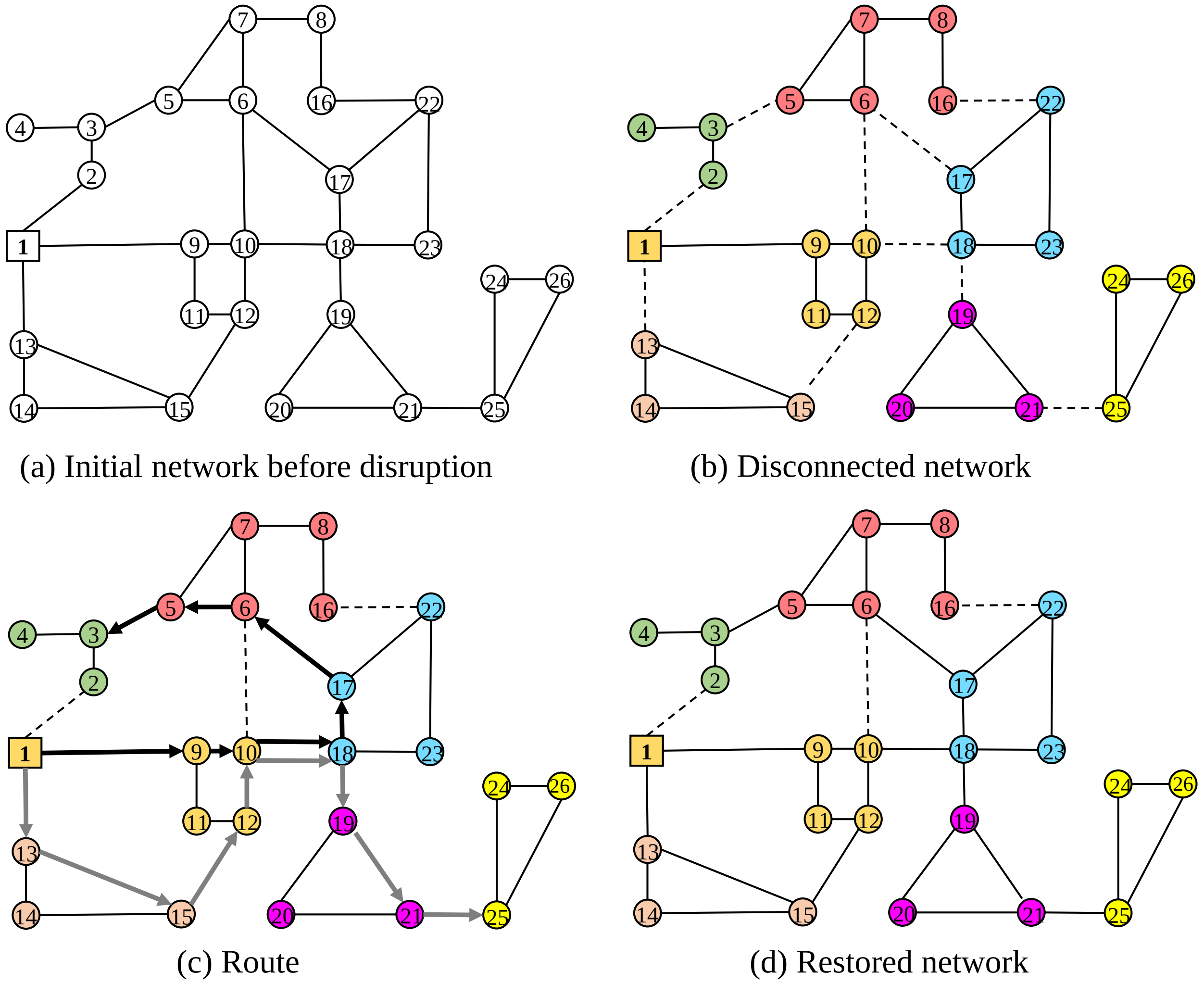
An example of this problem is given in Figure 1. Plot (a) in this figure gives the graph $G = (V, E)$ before the disruption. Plot (b) in this figure gives $G_B = (V, E \setminus B)$ which has 7 components in it. In this example we assume there are 2 road clearance crews that are initially positioned in node 1 as the depot. Plot (c) gives the paths of the clearing teams. First team goes from node 1 to 10 through node 9 and then opens (10, 18). Then from 18 goes to 17 and opens (6,17). Finally, it goes from 6 to 5 and opens (3,5). The second team, first opens (1,13) and then goes to 15 and opens (12,15). From 12 goes to 10 and uses (10,18) that has already been opened by team 1. This shows the coordination of the routes. After this point, team 2 opens (18, 19) and then goes to node 21 to finally opens (21, 25) when the network becomes reconnected. Plot (d) gives the restored network. As it was expected, it can be observed that we do not need to open all the links to ensure connectivity of the network.

4. Solution methodology

In this section we first explain the local search matheuristic algorithm suggested in Akbari & Salman (2017b) for K-ARCP. Then, we discuss the drawbacks of the proposed method and present our efficient algorithm called MIP-based heuristic with Rich Local Search (MIP-RLS) to handle more complex and bigger instances of this problem.

4.1. Local search heuristic in Akbari & Salman (2017b)

In Akbari & Salman (2017b) an exact MIP model is introduced to solve the K-ARCP. However, since the size of this exact model is very large, off-the-shelf optimizer packages cannot find a feasible solution within 3 hours even for moderate-sized instances (with 75 nodes). Thus, the authors have suggested a relaxation of



(a) Initial network before disruption

(b) Disconnected network

(c) Route

(d) Restored network

Figure 1: An example of the problem

the model, called R-MIP, where all the time-related constraints of the road clearance teams are relaxed. After solving R-MIP optimally, two possible scenarios can happen: 1) no timing conflicts exist in the outputted paths of the optimal solution. Hence, the optimal solution of the R-MIP is also the optimal solution of the K-ARCP; 2) timing conflicts exist in the outputted paths from its optimal solution. In this case, feasible solutions are obtained by changing the assignment of unblocking tasks to the road restoration teams and imposing the necessary waiting times. After the initial solution is extracted from the R-MIP, a local search algorithm based on swapping the opener of the blocked edges or swapping some parts of the paths of the vehicles with each other is developed to enhance the extracted initial solutions. The solution derived from solving the R-MIP provides a lower bound for the K-ARCP and the upper bound of this problem is obtained from the feasible solution extracted from the local search Algorithm.

Although this procedure shows quite good results in the tested instances of Akbari & Salman (2017b), it requires solving the R-MIP optimally. Several parameters from the input data set can affect the performance of the R-MIP significantly. In general, the number of vehicles (K), number of nodes ($|V|$), the amount of blocked edges ($|B|$) and the number of separated components (Q) can have contrariwise influence on the run time of the R-MIP. Figure 4 shows the results of testing R-MIP with 2 to 10 vehicles on instances generated from Istanbul road network. These two networks are the ”Simplified Istanbul” and the ”Southwestern” region of the Istanbul city that are used in Akbari & Salman (2017b) as well. Ten instances are generated and tested from each of these networks. The simplified Istanbul road network has 74 nodes and 179 edges (Figure 2) and the Southwestern Istanbul network has 250 nodes and 539 edges (Figure 3). The experiments are conducted on a computer with Intel Xeon W-2123 3.60 GHz 3.60 GHz processor, 32 GB RAM and 64-bit Windows 10 Professional operating system. The algorithms are coded in Python. We solve the MIP models using Gurobi Optimizer 9.0 in the Python environment.

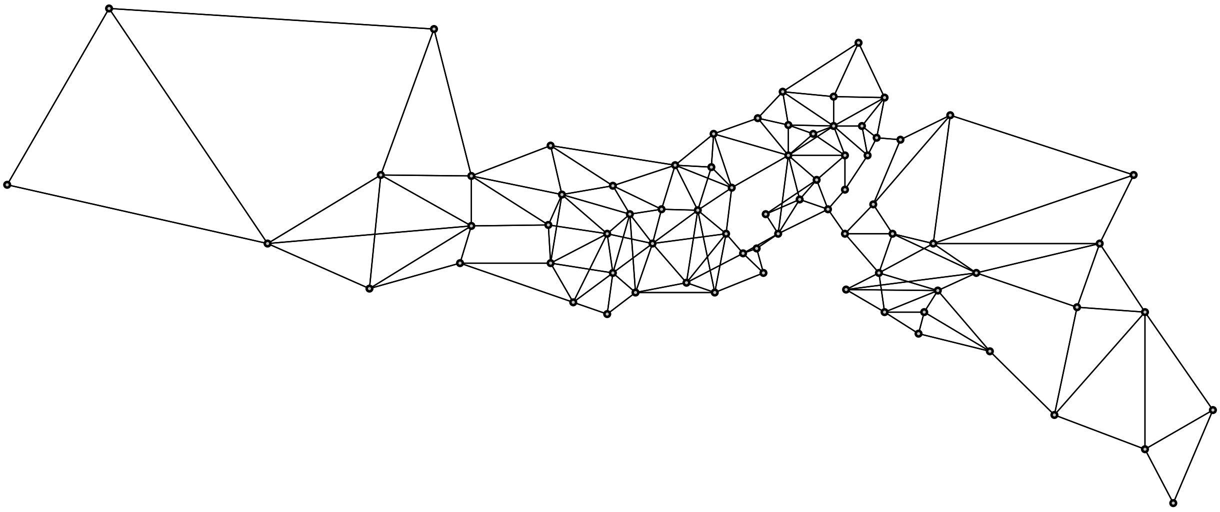


Figure 2: Simplified road network

The plots on Figure 4, (a) and (b), show the average performance of solving R-MIP for the Simplified Istanbul road network using 2 to 10 vehicles on 10 distinct instances over a 600 seconds time-limit. The average number of components in these 10 instances is 14.5 and the average number of blocked edges is 94.7 which is approximately 53% of all the edges. Plot (a) in Figure 4 shows the average percentage of the relative MIP gap for R-MIP over 600 seconds. This gap is extracted from the solver after 600 seconds. As the number of vehicles increases, the gaps increase in general and the solver is unable to solve real-sized instances of R-MIP in the given time-limit. The results of two instances with 16 and 12 separated components are given in plot (a) as well. For the instance with 16 components and more than 5 vehicles, the Gurobi solver could not find a feasible solution in 600 seconds in any of the cases. In the section (b) of Figure 4, “Solved optimally” shows the number of instances (out of 10) that has been solved optimally and “Not feasible” indicates the number of instances (out of 10) that the Gurobi solver could not find a feasible

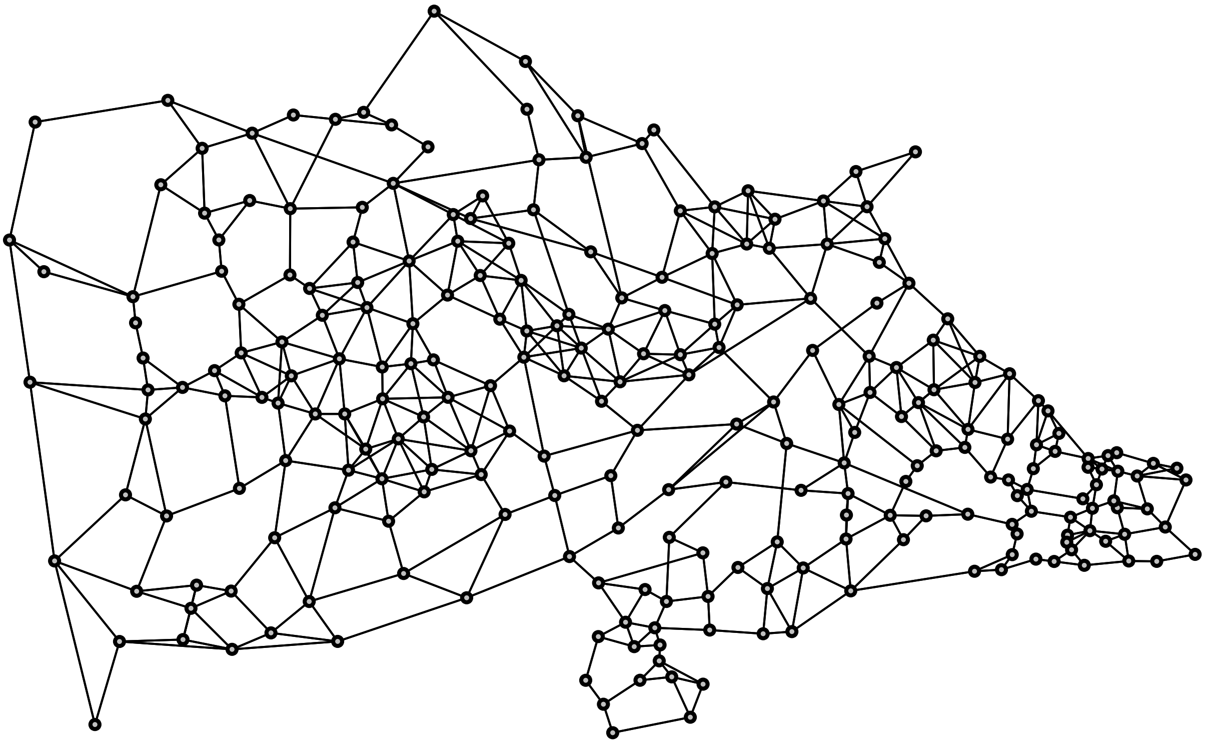


Figure 3: Southwestern road network

solution for them in 600 seconds. Note that in the algorithm developed in Akbari & Salman (2017b), the R-MIP should be solved optimally to obtain the lower bound and to test whether the extracted paths are feasible or not. Hence, if the R-MIP cannot be solved optimally, their algorithm cannot produce a solution to the K-ARCP. The results are shown using 2 to 10 vehicles. For instances with 5 vehicles, only 1 instance has been solved optimally in 600 seconds and in 2 of the tested instances gurobi was not able to find a feasible solution in 600 seconds. Plots (c) and (d) of Figure 4, show the same results on Southwestern Istanbul road network but in 1800 seconds time-limit as these networks are larger. On these instances, the average number of components is 15.1 and the average number of blocked edges is 187.1 which is approximately 35% of all the edges. Note that in the Southwestern instances of Akbari & Salman (2017b), the average number of blocked edges is less than 110 corresponding to less than 20% of all the edges. In plot (c) the average obtained relative mid-MIP gap using the R-MIP in 1800 seconds is given. These gaps are relatively large for a 30 minutes run time. Besides, since R-MIP is not solved optimally we cannot generate initial walks from its results. In Plot (c), the data of examples with 13 and 16 components are given as well. In the instance with 13 components, Gurobi did not find a feasible solution in 1800 seconds with 9 and 10 vehicles. In the given instance with 16 components, in none of the cases with more than 5 vehicles a feasible solution was found within 1800 seconds. Plot (d) in Figure 4 shows the number of instances solved optimally and the number of instances that a feasible solution could not be found in the given time-limit for the Southwestern instances. None of the instances with 5 and more vehicles have been solved optimally. As the number of road clearance teams increases in more cases the Gurobi solver fails to find a feasible solution in 30 minutes. It indicates that solving the R-MIP optimally is not a reliable method to solve K-ARCP for larger instances.

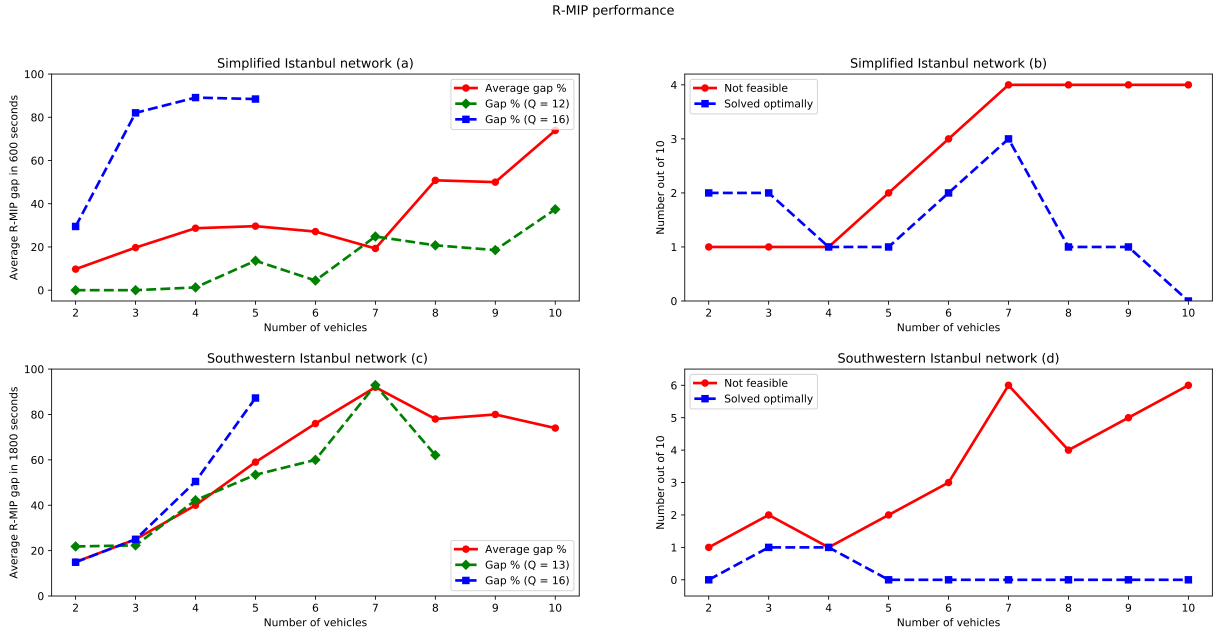


Figure 4: R-MIP performance analysis

4.2. *The MIP-based algorithm with Rich Local Search moves (MIP-RLS)*

The drawbacks of the developed matheuristic algorithm in Akbari & Salman (2017b) to solve K-ARCP are given in Section 4.1. We develop a new heuristic algorithm (MIP-RLS) that is capable of obtaining good feasible solutions in larger instances in acceptable time intervals (less than 20 minutes). In the following sections, first the initial solution generation procedure and then our proposed problem-specific rich neighborhood moves of the MIP-RLS are given.

4.2.1. *Initial solution generation for the MIP-RLS*

In MIP-RLS, instead of generating the initial solutions using common procedures such as Greedy Randomized Adaptive Search Procedure (GRASP), we develop a novel heuristic and obtain the initial solutions by solving a binary problem. We will refer to this problem as the Initial Solution Binary Problem or the ISBP. Different from the R-MIP that is a step-based formulation with numerous binary and integer variables and several step-related constraints, we pre-process the problem to identify two sets of parameters that enable us to develop a path-based formulation with considerably less constraints and variables.

We denote the first parameter set with $T_q : q = \{1, 2, ..., Q\}$ which shows the *earliest time* that each component q can be opened. In order to obtain T_q, the minimum of shortest path distances from depot to all the nodes in V_q should be calculated. These shortest paths should be calculated considering the blockage times. In order to reflect this observation we define a transformation of the graph $G_B = (V, E \setminus B)$ into another graph $T(G_B) = (V, E)$ such that V and E are the set of all nodes and all edges, respectively. The difference between $T(G_B)$ and $G(V, E)$ is that the traversal time of blocked edges B, is set to its traversing time plus the opening time, i.e. $c_{ij}+b_{ij}$. Moreover, there is a path R_q associated with T_q which is the shortest path from the depot to a node in V_q on $T(G_B)$. The second set of parameters that are calculated in the pre-processing step are denoted by $C_q^{q'} \in \{0, 1\} : q, q' = \{1, 2, ..., Q\}$. This set of parameters are defined to bundle components that opening them requires to open another set of components. $C_q^{q'}$ is a *binary* parameter that equals 1 if component q' has been opened in the calculated shortest path of component q; and equals 0, otherwise i.e., there are nodes from q' in R_q. Due to frequent usage of the word "bundle" in this study, we will clearly define it in the following. Components q and q' are in the same bundle if $C_q^{q'} = 1$ or $C_{q'}^{q} = 1$. An example of such relation is shown in Figure 5. In Figure 5, R_2 depicts the shortest path connecting component 2 to the depot. On this path, the vehicle should visit a node in component 3 before traversing to component 2 which implies that $C_2^3 = 1$ and therefore, components 2 and 3 are in the same bundle.

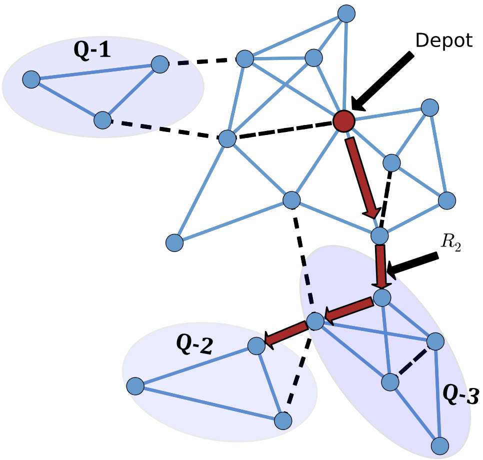


Figure 5: $C_q^{q'}$ Parameters demonstration

Given parameters T_q and $C_q^{q'}$, in what follows, we first provide the definition of the decision variable and then, define constraints of ISBP:

Decision variables of ISBP:

$$z_q^k = \begin{cases} 1, & \text{if path } R_q \text{ is assigned to vehicle } k = 1, 2, ..., K,\ q = 1, ..., Q \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$w : \text{An intermediate continuous decision variable indicating the length of the longest path} \tag{2}$$

ISBP:

$$Min \quad w \tag{3}$$

$$w \geq \sum_{q=1}^{Q} T_q z_q^k, \quad k = 1, 2, \ldots, K \tag{4}$$

$$\sum_{k=1}^{K} \sum_{q=1}^{Q} C_q^{q'} z_q^k \geq 1, \quad q' = 1, 2, ..., Q. \tag{5}$$

$$z_q^k \in \{0, 1\}; k \in \{1, ..., K\}, q \in 1, ..., Q \tag{6}$$

$$w \geq 0 \tag{7}$$

Objective function shown by (3) and Constraints (4) set the objective function that minimizes the longest path. Note that more than one T_q can be assigned to a vehicle and by $\sum_{q=1}^{Q} T_q z_q^k$ the path length of vehicle k can be calculated. Constraints (5) ensure that all components get connected by assigning all components to at least one vehicle. $C_q^{q'} z_q^k$ will equal 1 for component q' only if both $C_q^{q'}$ and z_q^k equal 1. That is, when path R_q includes opening component q' and its opening procedure is assigned to vehicle k. In here, we point out that while each component should be assigned to at least one vehicle, a solution to the ISBP, does not satisfy $\sum_{k=1}^{K} z_q^k = 1, q \in \{1, ..., Q\}$. It is because some of the components that are in the same bundle, might be assigned to the same vehicle. Moreover, z_q^k does not indicate whether vehicle k visits component q or not, and it shows whether path P_q is assigned to vehicle k or not. Different from the R-MIP which is a computationally difficult optimization model, the ISBP is an easy binary problem and it can be solved in a short time (less than 1 minute) within the content of its application in our study.

We extract the optimal decision variable values of ISBP once it is solved optimally and denote them by z_q^{*k}. Considering this solution, we can identify the set Q_k for each vehicle $k \in \{1, ..., K\}$, so that it contains the components that should be opened by vehicle k. In mathematical notations; $Q_k = \{q' \in \{1, ..., Q\} : \sum_{q=1}^{Q} C_q^{q'} z_q^{*k} = 1\}$. With Q_k in hand, we can solve an MIP model to obtain a path for vehicle k, such that the path length is minimized and it covers all the components in Q_k. We refer to this MIP model, the "Single-Vehicle Component Coverage" and denote it by SVCC. The mathematical formulation for the SVCC problem is given in subsection 4.2.2. Since in the single-vehicle instances of the K-ARCP, all the components should be opened while in the SVCC only the components that are in Q_k should be opened, it is a simpler version of the ARCP that should be solved in a subset of components of the entire network. Hence, each time that the

SVCC is solved for a set of components, we eliminate the parts of the graph, i.e. the components that should not be opened and traversed. Although the computational tests on SVCC demonstrate that it can be solved in less than 2 minutes in all the tested instances, we define Procedure 1 to avoid resolving similar instances.

Procedure 1: SVCC

```
procedure SVCC($\hat{Q}$)
                                        ▷ The SVCC problem for all the components in $\hat{Q}$.
    if $\hat{Q} \in D_{temp}$ :   then   return   $temp(\hat{Q})$          ▷ Possible termination point.
    else
        $D_{temp} = D_{temp} \cup \hat{Q}$
        $temp(\hat{Q})$ = optimal objective value of the SVCC with set $\hat{Q}$
        return   $temp(\hat{Q})$
    end if
end procedure
```

The input of the SVCC procedure is a set of components, $\hat{Q}$ and the output is the minimum tour length associated with opening them, $SVCC(\hat{Q})$. In Procedure 1, we define a one-to-one function called *temp* whose domain is denoted by D_{temp}. Initially, D_{temp} is an empty set ($D_{temp} = \emptyset$). If the optimal path for traversing a set of components is calculated, the set of components will be added to the domain of *temp* and the $temp(\hat{Q})$ will be the tour length associated with opening those components. Using the *temp* function, we avoid recalculating the optimal tour for a set of components that is already calculated.

Having solved SVCC for all the vehicles $k \in \{1, ..., K\}$ using the set Q_k, we can derive a path for each vehicle. Let us denote these paths by $\mathcal{P}_k$. In the following we show two propositions; in the first one we show that we can obtain a feasible solution from the obtained solutions of SVCC problems, in the second one we claim that the collection of $\mathcal{P}_k$ gives the optimal solution to the K-ARCP in special cases.

Proposition 1. *The collection of $\mathcal{P}_k$ for all the vehicles $k = 1, ..., K$, gives a feasible solution to K-ARCP.*

Proof. For these paths to be feasible two conditions should hold; 1) the collection of the paths should open all the components, 2) paths should be coordinated, meaning that no blocked edges is traversed unless it is opened. First condition holds because of the Constraints (5) given in ISBP. For the second condition, we should notice that for each vehicle k, $\mathcal{P}_k$ corresponds to a set of intact and blocked edges that need to be traversed. In these paths, traversal of a blocked edge always results in the additional unblocking time and hence the timings cannot be infeasible. □

Proposition 2. *Let us denote the optimal objective function value of the ISBP and the K-ARCP with Z^*_{ISBP} and Z^*_{K-ARCP}, respectively. If $Z^*_{ISBP} = \max_{q=1}^{Q} T_q$ then the derived paths from ISBP, $\mathcal{P}_k$, are the optimal paths to K-ARCP, i.e. $Z^*_{ISBP} = Z^*_{K-ARCP}$.*

Proof. Let us distinguish component $\hat{q}$ such that $T_{\hat{q}} = \max\limits_{q=1}^{Q} T_q$. As by definition of parameter T_q, the minimum time required to reconnect component $\hat{q}$ is $T_{\hat{q}}$, the minimum time to reconnect all the components should be higher than $T_{\hat{q}}$, $Z^*_{K-ARCP} \geq T_{\hat{q}}$. Since $\mathcal{P}_k$ gives paths that reconnect all the components in $T_{\hat{q}}$ and by Proposition 1 the collection of these paths is a feasible solution to K-ARCP hence, $Z^*_{K-ARCP} \leq T_{\hat{q}}$. It implies that $Z^*_{K-ARCP} = T_{\hat{q}}$ and $Z^*_{ISBP} = Z^*_{K-ARCP}$. □

Corollary 1. *A collection of paths that provides a feasible solution to the K-ARCP and has an objective function value equal to* $\max\limits_{q=1}^{Q} T_q$ *gives an optimal solution to the K-ARCP.*

Proof. The proof follows with the same argument in the proof of Proposition 2. □

In the following we define a procedure that calculates the objective function value that corresponds to a component assignment to all vehicles, $Q = \bigcup\limits_{k=1}^{K} Q_k$. Recall that for each road clearance team $k \in \{1, ..., K\}$ a set Q_k shows the components that vehicle k opens them and $\mathcal{P}_k$ gives the path associated with that. Furthermore, let us assume ℓ_k denotes the length of path $\mathcal{P}_k$.

Procedure 2: objective function of a component assignment Q

```
procedure ObjCalc($Q$)
    for $k \in \{1, ..., K\}$ do
        $\ell_k \leftarrow SVCC(Q_k)$
    end for
return $\max\limits_{k} \ell_k$
end procedure
```

In procedure 2, first we calculate the path length associated with each vehicle considering the component set that is assigned to it using Procedure 1. Then, we set the corresponding objective value of Q to the longest path among all the vehicles i.e. the makespan.

In the following we define another procedure that shows whether an assignment of components to vehicles corresponds to an optimal solution of K-ARCP or not. In Procedure 3, the input is a set Q with K subsets, Q_k, that shows the assigned components to vehicle $k = 1, ..., K$. The output is either True or False. It is True if by using Corollary 1 we can show that Q is an optimal component assignment to vehicles and False, otherwise. We use Procedure 2 to determine the objective function value corresponding to Q. If it is equal to the largest $T_q, q \in \{1, ..., Q\}$, then Q corresponds to an optimal solution and the output of OptCom(Q) will be True. The output will be false otherwise.

Procedure 3: optimality of component assignments to vehicles

```
procedure OptCom($Q$)
    if ObjCalc($Q$) = $\max_{q=1}^{Q} T_q$ then return True
    else return False
    end if
end procedure
```

4.2.2. The Single-Vehicle Component Coverage problem

The Single-Vehicle Component Coverage (SVCC) problem can be defined as finding the best path for a vehicle k to open all the components in the set Q_k. Recall that V_q and E_q give the set of nodes and edges for component $C_q, q \in \{1, ..., Q\}$, respectively. Furthermore, in numbering the components, we assume that the depot is located in component 1. In order to remove non optimal feasible solutions, we define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of all nodes in the component with the depot and all the nodes which are in the components that should be opened by the corresponding vehicle, $\mathcal{V} = \{i : i \in V_q, C_q \in Q_k\} \cup V_1$; and $\mathcal{E}$ is the set of all intact edges in the union of C_1 and Q_k, $\mathcal{E} = \{(i, j) : (i, j) \in E_q, C_q \in Q_k\} \cup E_1 \setminus \mathcal{B}$ where $\mathcal{B}$ is the set of all blocked edges in component 1 and in all components q such that $C_q \in Q_k$. Other parameters such as the traversal time of edges or opening times of blocked edges will remain the same as in the K-ARCP for the SVCC. In the following, first provide the definition of the decision variables and then show the formulation.

Table 1: Definition of the decision variables utilized in the SVCC

Sign	Restrictions	Description
x_{ij}, $\forall (i, j) \in A$	$\{0, 1\}$	1 if (i, j) is traversed in direction from i to j; 0, ow.
z_{ij}, $\forall (i, j) \in \bar{\mathcal{B}}$	$\{0, 1\}$	1 if (i, j) is unblocked; 0, ow.
f_{ij}, $\forall (i, j) \in A$	≥ 0	The amount of flow in the direction from node i to j.
v_i,$i \in \mathcal{V} \cup \{n + 1\}$	≥ 0 (Integrality Relaxed)	Number of times node i is visited by the road clearance team.

$$Min \sum_{(i,j)\in A} c_{ij} x_{ij} + \sum_{(i,j)\in \bar{\mathcal{B}}} b_{ij} z_{ij} \tag{8}$$

$$\sum_{j:(D,j)\in A} (x_{Dj} - x_{jD}) = 1 \tag{9}$$

$$\sum_{j:(i,j)\in A, j\neq n+1} (x_{ij} - x_{ji}) = 0, \quad \forall i \in \mathcal{V} \backslash D \tag{10}$$

$$\sum_{j\in \mathcal{V}} x_{j(n+1)} = 1 \tag{11}$$

$$x_{ij} \geq z_{ij}, \quad \forall (i, j) \in \bar{\mathcal{B}} \tag{12}$$

$$x_{ij} + x_{ji} \leq 2(z_{ij} + z_{ji}), \quad \forall (i, j) \in \bar{\mathcal{B}} \tag{13}$$

$$\sum_{j:(j,i)\in A} x_{ji} = v_i, \quad \forall i \in \mathcal{V} \tag{14}$$

$$\sum_{i \in V_q} v_i \geq 1, \quad q : C_q \in Q_k \tag{15}$$

$$\sum_{j:\in \mathcal{V} \cup \{(n+1)\}:(i,j)\in A} (f_{ij} - f_{ji}) = -v_i, \quad \forall i \in \mathcal{V} \cup \{(n+1)\} \setminus D \tag{16}$$

$$\sum_{j \in \mathcal{V} \cup \{(n+1)\}} (f_{Dj} - f_{jD}) = \sum_{i \in \mathcal{V} \cup \{(n+1)\} \setminus \{D\}} v_i, \tag{17}$$

$$\sum_{j \in \mathcal{V}} f_{j(n+1)} = 1 \tag{18}$$

$$f_{ij} \leq |\mathcal{V}| x_{ij}, \quad \forall (i, j) \in A \tag{19}$$

$$f_{ij} \geq x_{ij}, \quad \forall (i, j) \in A \tag{20}$$

$$x_{ij} \in \{0, 1\}; (i, j) \in A \tag{21}$$

$$z_{ij} \in \{0, 1\}; (i, j) \in \bar{\mathcal{B}} \tag{22}$$

$$f_{ij} \geq 0; (i, j) \in A \tag{23}$$

$$v_i \geq 0; i \in \cup \{n + 1\} \tag{24}$$

The objective function given in (8) minimizes the the total unblocking and traversal times. Constraints (9) to (11) are the vehicle balance equations. By constraint (9) the vehicle starts its path from the depot. By constraints set (10), once the vehicle enters a node other than the depot or the sink node, should also leave it. By constraint (11) the path of the vehicle should ends in the sink node. By constraint (12) a blocked edge is only opened if the road clearance team traverses it and by constraint (13) the traversal of a blocked edge is only viable after it is unblocked. Constraint (14) the number of times the road clearance teams visits node $i \in \mathcal{V}$ is counted. By constraints (15), at least one of the nodes in each of the components that are assigned to vehicle k should be visited. Note that this formulation is given for vehicle k. Constraints (16) to (20) are defined to guarantee the connectivity of the walks. Detailed description of these constraints together with an illustrative example can be found in Akbari & Salman (2017a). The remaining constraints from (21) to (24) are variable restrictions that are defined in Table 1.

4.2.3. Rich Local Search procedure

After obtaining $\mathcal{P}_k$ from the solutions of the SVCC problems, we can initiate the Rich Local Search (RLS) procedure. Since we obtain the path of each vehicle using the SVCC procedure, we cannot further improve them and we do not have any inter-path local search moves. Hence, new intra-path moves should be defined to interrelate the paths of different vehicles. An important difference between our developed RLS moves and the local search in the literature is that instead of changing the order of visiting nodes, which is a common procedure for most of the local search moves in the literature, we update the components assigned to vehicles to obtain better objective functions. Moreover, our RLS moves are also different from the ones in Akbari & Salman (2017b) which develop inter and intra local search moves. The intra moves include swapping blocked edges or parts of the paths between vehicles when they traverse the same blocked edges. However, our RLS moves, changes the assignment of a whole component to a vehicle and hence considers larger neighborhoods. As it is stated above, it does not require inter-path moves as the optimal path for covering the corresponding components is obtained by solving the SVCC model.

Furthermore, our RLS moves consider that only shortening the length of the longest path might result in a better objective function value. In the following we define some notations and then these new and rich problem specific moves are explained.

Once the initial paths associated to each vehicle are extracted from the ISBP model, without loss of generality, let us sort the index of the vehicles based on the length of the paths denoted ℓ_k values such that $\ell_k \geq \ell_{k+1}, k \in \{1, ..., K-1\}$. For each component q in Q_k let N_q^k denote the number of components bundled with component q in the path of vehicle k. This value is calculated as it is shown in equation (25) and its value is 0 for those component that their opening process does not involve traversing other components and they are not required to be traversed to open other components.

$$N_q^k = \begin{cases} \text{if } \; z_q^{*k} = 1 : & \sum\limits_{q'=1, q' \neq q}^{Q} (C_q^{q'} z_q^{*k}) \\ \text{if } \; z_q^{*k} = 0 : & \sum\limits_{q'=1, q' \neq q}^{Q} (C_{\hat{q}}^{q'} z_{\hat{q}}^{*k}), \quad \hat{q} \in \{1, ..., Q\} \text{ such that: } C_{\hat{q}}^{q} z_{\hat{q}}^{*k} = 1 \end{cases} \tag{25}$$

Below we give the novel rich local search moves developed for the K-ARCP. These moves are defined to balance the paths obtained by SVCC problems. In all of these moves the input is the set of components assigned to each vehicle (Q_k) and the corresponding path length to cover these components (ℓ_k). The output is a new set of components assigned to each vehicle. Note that different from regular local search moves used in routing problems, here we do not change the node assignments to vehicles, but we change the assignment of components to the vehicles and re-optimize their paths. Furthermore, we do not change the order of visiting the components in these moves as the SVCC model decides about that.

- **Remove insert:** By this operator, one of the assigned components to vehicle 1 is moved to Q_K. In other words we eliminate a component q from Q_1 and add it to the components that are assigned to vehicle K, Q_K. Recall that before this move vehicle 1 has the longest path and vehicle K has the shortest path

($\ell_k \geq \ell_{k+1}, k \in \{1, ..., K-1\}$). The selection process is not random, we pick component q in Q_1 such that $N_q^1 = 0$. If none of the components satisfy this condition, we skip this move.

- **Swap with shortest:** Swap with Shortest is defined to elaborate *Remove insert* in instances that it is not applicable. By this move, we swap one or more of the assigned components between vehicle 1 and vehicle K. In other words, we pick the smallest bundle from Q_1 and swap it with the smallest bundle in Q_K. To pick a component with the smallest bundle, for instance for vehicle 1, we need to pick all components $\check{q}$ such that $N_{\check{q}}^1 \leq N_q^1$.

- **Double swap:** This swap is defined not only to consider the longest and shortest paths but also to consider the second largest and second shortest paths. In this move, we use the same *Swap with shortest* operator but with the difference that it will be done once between Q_1 and Q_K and once between Q_2 and Q_{K-1}. Similarly, we pick the smallest bundles from the paths of each vehicle.

- **Top double swap:** This swap is useful when ℓ_1 is considerably larger than other paths. In this move, we pick two of the smallest bundles from Q_1 and move them to Q_K and Q_{K-1}.

- **Perturbation:** By this operator, we first pick the smallest bundles for vehicles 1 to K. Next, we move the chosen bundle from Q_k to Q_{k+1} for $k = 1, ..., K-1$. Using this move all the paths are subject to changes. Perturbation is added as an step to avoid local optima.

Note that each time that the assignment of the components to the vehicles changes, paths of all the vehicles, $\mathcal{P}_k, k \in \{1, ..., K\}$ should be recalculated to be feasible. Procedure 1 is defined to avoid recalculating the paths corresponding to the same component assignments.

Using the initial solution generation procedure from Section 4.2.1 and the RLS moves defined in 4.2.3 the MIP-RLS algorithm can be shown as in Procedure 4.

On line 1 of the MIP-RLS the graph $T(G_B) = (V, E)$ is formed by setting the traversal time of blocked edges to be equal to $c_{ij} + b_{ij}$. On line 2, the parameters T_q and $C_q^{q'}$ are calculated based on the input information. On line 3, we solve the Initial Solution Binary Problem (ISBP) with the calculated parameters, T_q and $C_q^{q'}$ and then set Z_q^{*k} values to be equal to the optimal solution of it. On lines from 4 to 7, using the Z_q^{*k} values, first for each vehicle k, we calculate $Q_k = \{q' \in \{1, ..., Q\} : \sum_{q=1}^{Q} C_q^{q'} z_q^{*k} = 1\}$ and then using the set Q_k and the SVCC, we can obtain path $\mathcal{P}_k$. By line 9, if Proposition 2 and the OptCom Procedure show that Q gives the optimal component assignment to the K-ARCP, the algorithm terminates and otherwise we use this solution as the initial solution and start the improvement phase using our rich local search moves proposed in section 4.2.3. We showed all the Possible Termination Points in the algorithm with PTP. The

Procedure 4: MIP-based heuristic with Rich Local Search for the K-ARCP (MIP-RLS)

```
Input:
     All the parameters of the input problem.          ▹ (Sets $V$, $E$, $B$, $c_{ij}$, $b_{ij}$, $Q$, $K$ and $Depot$).
 1: Form $T(G_B)$ which is a transformation of $G(V, E)$.
 2: Calculate parameters $T_q$ and $C_q^{q'}$.
 3: $Z_q^{*k} \leftarrow ISBP$
 4: for $k \in \{1, ..., K\}$ do
 5:     $Q_k \leftarrow Z_q^{*k}$
 6:     $\mathcal{P}_k \leftarrow SVCC(Q_k)$
 7: end for
 8: $Q = \bigcup_{k=1}^{K} Q_k$
 9: if OptCom($Q$) = $True$ then :   return $Q$   end if          ▹ PTP
10: while "Time-limit" do
11:     $Q_{new}$ = $Remove\ insert\ (Q,\ell)$
12:     if ObjCalc($Q_{new}$) < ObjCalc($Q$) then :   $Q = Q_{new}$
13:         if OptCom($Q$) = $True$ then :   return $Q$          ▹ PTP
14:         end if
15:     end if
16:     $Q_{new}$ = $Swap\ with\ shortest\ (Q,\ell)$
17:     if ObjCalc($Q_{new}$) < ObjCalc($Q$) then :   $Q = Q_{new}$
18:         if OptCom($Q$) = $True$ then :   return $Q$          ▹ PTP
19:         end if
20:     end if
21:     $Q_{new}$ = $Double\ swap\ (Q,\ell)$
22:     if ObjCalc($Q_{new}$) < ObjCalc($Q$) then :   $Q = Q_{new}$
23:         if OptCom($Q$) = $True$ then :   return $Q$          ▹ PTP
24:         end if
25:     end if
26:     $Q_{new}$ = $Top\ double\ swap\ (Q,\ell)$
27:     if ObjCalc($Q_{new}$) < ObjCalc($Q$) then :   $Q = Q_{new}$
28:         if OptCom($Q$) = $True$ then :   return $Q$          ▹ PTP
29:         end if
30:     end if
31:     $Q$ = $Perturbation\ (Q,\ell)$
32: end while
33: return $Q$          ▹ PTP
```

lines from 10 to 32 gives the application of the defined RLS moves. The termination condition of the MIP-RLS is a time-limit which is set in line 10. In our application, we set the time-limit to be at most 20 minutes. However, the algorithm can end before the time-limit is reached by finding the optimal solution in any of the possible termination points. In lines 11, 16, 21 and 26 an updated assignment of components to vehicles is created based on the defined search moves. The input of these search moves are the current assignment of components and the length of the associated paths for all vehicles. This updated assignment of components is called Q_{new}. On lines 12, 17, 22 and 27 if the corresponding objective function of the created component assignment set Q_{new} using ObjCalc procedure gives a better objective function compared to the current Q, the best component assignment will be updated. On lines 13, 18, 23 and 28 optimality of the current best assignment is checked using the OptCom procedure and if the current solution is optimal, the algorithm terminates. On line 31 a perturbation move is implemented to perturb the assignments of the components to the vehicles. This step update the sets Q_k in all the cases to skip local optimal solutions. If the time-limit is reached, meaning that an approve-able optimal solution is not found, the algorithm exits the search process. In line 33 the best component assignment set is outputted. The path of each vehicle can be determined using the SVCC model.

5. Data and computational study

To experiment the performance of the MIP-RLS we used two networks from Istanbul city that are introduced in Akbari & Salman (2017b). The first road network that we consider, is a Simplified road network from Istanbul in which only 74 nodes and 179 edges are considered. The second road network however, is a detailed road network obtained from the Southwestern region of Istanbul. In this detailed road network of the Southwestern Istanbul, 250 nodes and 539 edges are considered. Even though these road networks were exactly the same as those of the literature, we have tested more intense scenarios with more blocked edges as we are targeting larger data sets in terms of the intensity of the disaster and hence the amount of blocked edges and separated components. For example in the Southwestern instances tested in Akbari & Salman (2017b), in all the instances under 20% of the edges were blocked but in our study, on average, approximately 35% of the edges are assumed to be blocked. The number of these blocked edges has a direct impact on the number of separated components that determines the difficulty of the problem as well. In the following, we first discuss the optimality gap of the MIP-RLS and next, give the results of the tested instances.

5.1. On the optimality gap of the MIP-RLS

Before giving the results on the larger instances, an analysis on the performance of the MIP-RLS is necessary. Since the objective function is to minimize the time of the longest road clearance team route, the obtained feasible walks will give an upper bound on the optimal solution value. However, in order to get a lower bound, as it is suggested in Akbari & Salman (2017b) we need to solve R-MIP optimally which is not possible as R-MIP cannot be solved optimally in our intended instances within a reasonable time-limit. The weak performance of the R-MIP in larger instances is also discussed in Section 4.1. Instead, we generated 10 smaller instances of the Simplified Istanbul network and tested them with both R-MIP and MIP-RLS by 2 to 4 vehicles. These instances were small enough to be solved by R-MIP in a one hour time-limit. On these instances the average value of Q is 8.3 varying from 7 to 10 and on average 77.7 edges are blocked. For instances 1 and 2, $Q = 7$, for instances 3 to 6, $Q = 8$ for instance 10, $Q = 10$ and for the remaining instances this value is equal to 9.

Figure 6 illustrates the results of the tested instances and Table 2 presents a summary of the 10 tested instances with 2 to 4 vehicles. Plot (a) in Figure 6 shows the optimality gap in the tested instances of MIP-RLS. This gap is calculated by setting the results of the MIP-RLS as the upper bound and the results of the R-MIP as the lower bound. All the instances with 4 vehicles have been solved optimally. Among 10 instances, 7 of them with 3 vehicles have been solved optimally and among those which have not been solved optimally the maximum optimality gap is 7.11%. In instances with 2 vehicles 4 instances have been solved optimally and on the other instances the average optimality gap is 2.44%. In Figure 6 plots (b), (c) and (d) show the run time comparison of the tested instances between solving R-MIP and MIP-RLS with 2,3 and 4 vehicles respectively. Note that in these graphs, the MIP-RLS time is calculated by adding the initiation and search steps of the algorithm. Only in instances 5 and 6 with two vehicles the execution time of R-MIP is less than that of MIP-RLS. In all other instances the run time of MIP-RLS is significantly shorter than that of R-MIP, particularly when the number of the road clearance teams increases.

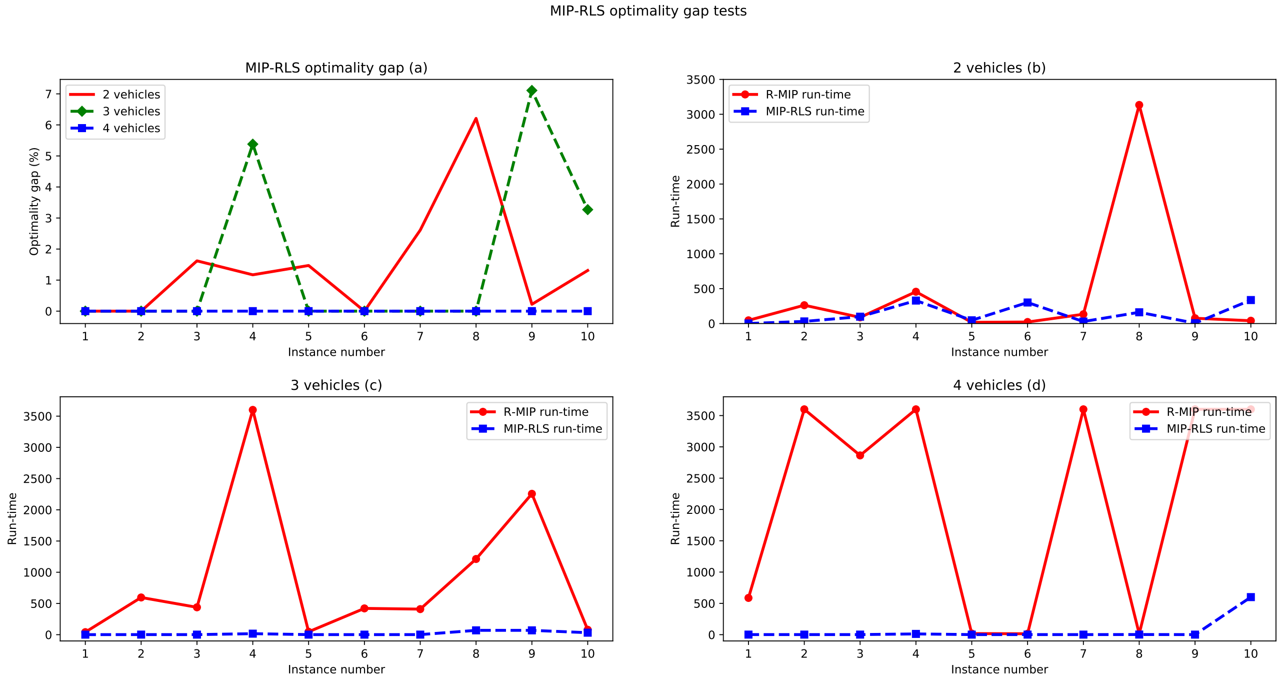


Figure 6: MIP-RLS Optimality gap analysis

Table 2 summarizes the results of these tested instances. Column $|\bar{Q}|$ gives the average number of separated components, $|\bar{B}|$ gives the average number of blocked edges, K is the number of road clearance teams, R-MIP Average Run time and MIP-RLS Average Run time columns provide the corresponding values respectively. As mentioned, these given times for the MIP-RLS algorithm include both the initiation and search steps. The last column gives the achieved average optimality gap for the MIP-RLS over the 10 instances. Note that these instances are small-sized networks and are tested for benchmarking and verifying the performance of the MIP-RLS and their optimality gap is calculated by setting the solution from the MIP-RLS as the upper bound and the results of the R-MIP as the lower bound. While the R-MIP average run time increases as the number of vehicles increases, the MIP-RLS found the optimal solution of all the tested instances with 4 vehicles on average in 62.26 seconds which is 2.89% of the required run time to solve the R-MIP for the same instances. For instances with 2 and 3 vehicles, the MIP-RLS shows good performance as well. While the average run time is less than 3 minutes for instances with 2 vehicles, the average optimality gap is less than 1.5%. For instances with 3 vehicles the run time decreases to less than 20 seconds while the average optimality gap remains under 1.6%.

Table 2: MIP-RLS optimality test

$\lvert\bar{Q}\rvert$	$\lvert\bar{B}\rvert$	K	R-MIP Average Run time	MIP-RLS Average Run time	MIP-RLS Average Optimality Gap
8.3	155.4	2	427.68	134.04	1.46%
		3	910.02	19.43	1.58%
		4	2149.55	62.26	0.00%

5.2. *Numerical results on large-sized K-ARCP instances*

We have tested 10 instances on each of the Simplified Istanbul and Southwestern networks using 2 to 10 vehicles. Note that in the largest instances of Akbari & Salman (2017b), four vehicles are used for Istanbul road networks whilst we test our algorithm with up to 10 vehicles. The average number of separated components, Q, in our tested instances is 14.5 and 15.1 for Simplified Istanbul and Southwestern instances, respectively, while in the tested instances of K-ARCP in Akbari & Salman (2017b), the average values of Q over 15 tested instances are 5.8 and 4.6 for Simplified Istanbul and Southwestern Istanbul instances, respectively. It shows that the instances tested in this study are larger than those of the literature. The average number of blocked edges in the tested instances in our study for Simplified Istanbul is 94.7 and for Southwestern Istanbul is 186.1. Comparing to literature on K-ARCP where the average number of blocked edges on Simplified Istanbul instances is 72.3, and for Southwestern Istanbul is 108.2, in this study, we focus on larger instances in this aspect as well.

In Section 4.1 we provided some results on the performance of R-MIP on the same instances. Although R-MIP could not be solved optimally in most of these instances in the given time-limit, in the following we will show that MIP-RLS solves many of the instances in a reasonable time-limit. The results of testing the MIP-RLS and R-MIP algorithms on the large-sized Simplified and Southwestern instances are given in Tables 3 and 4 for the Simplified and the Southwestern networks, respectively. In these tables, columns “ins”,

”$|Q|$”. ”$|B|$” and “$|K|$” respectively indicate the instance number, number of components, number of blocked edges and number of used vehicles. The optimally gap of the R-MIP in the given time-limit, Best Objective Function Value (OFV) obtained by the MIP-RLS and the run time for each instance in seconds are reported in the next three columns. In the instances for which the R-MIP Gap (%) is shown by —, the Gurobi solver could not find a feasible solution in the given time-limit. In Tables 3 and 4, the optimal solutions obtained by the MIP-RLS are **bolded**. The optimality of these solutions is confirmed by Proposition 2.

Figure 7 visualizes the run time results of the R-MIP and the MIP-RLS on the Simplified and Southwestern instances. The run time of the MIP-RLS takes into account both the initiation and search steps. In the plot (a), the simplified instances are tested over 600 seconds time-limit. The run time of the R-MIP shows an increasing trend as the number of vehicles increases. Note that as K increases, R-MIP hits the time-limit without finding an optimal solution in most of the instances. This shows that we are not able find a feasible solution for this instances using the algorithm proposed in Akbari & Salman (2017b). However, our MIP-RLS solves several instances optimally and the run time is considerably shorter. As the number of road clearance teams increases, the MIP-RLS have shorter run times and terminates faster by finding the optimal solution. Plot (b) of Figure 7 provides and illustration for the results of the instances on the Southwestern Istanbul network where the time-limit is set to be 1800 seconds for R-MIP and 1200 seconds for MIP-RLS. While the R-MIP could not be solved and hit the time-limit in almost all of the instances, the MIP-RLS finds a feasible solution in a significantly shorter time and it finds the optimal solution to most of the tested instances particularly when the number of road clearance teams increases. Again, it can be seen that in general the run time of the R-MIP is increasing in the number of vehicles. In contrast, when the number of vehicles increases the MIP-RLS run time decreases and it terminates by finding the optimal solution to most of the tested instances.

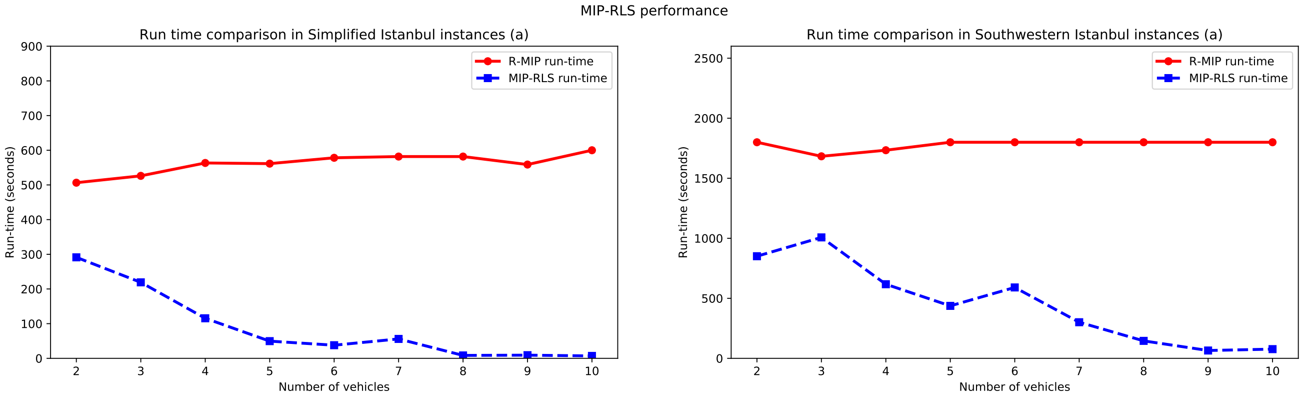


Figure 7: MIP-RLS run time analysis

Figure 8 illustrates the number of instances solved optimally among the tested instances by MIP-RLS.

Table 3: Results for large-size Simplified K-ARCP instances

ins	$\lvert Q\rvert$	$\lvert B\rvert$	$\lvert K\rvert$	R-MIP	MIP-RLS		ins	$\lvert Q\rvert$	$\lvert B\rvert$	$\lvert K\rvert$	R-MIP	MIP-RLS	
				GAP (%)	Best OFV	Time (s)					GAP (%)	Best OFV	Time (s)
1	16	186	2	29.48%	13.49	600	6	13	188	2	3.46%	21.72	600
			3	82.05%	10.58	600				3	0.77%	16.81	600
			4	89.11%	8.91	600				4	5.16%	**15.8**	6.39
			5	88.44%	7.75	600				5	0.51%	**15.8**	6.43
			6	—	6.58	600				6	0.36%	**15.8**	2.17
			7	—	**6.33**	11.16				7	1.80%	**15.8**	5.41
			8	—	**6.33**	4.06				8	0.00%	**15.8**	5.48
			9	—	**6.33**	3.73				9	0.46%	**15.8**	6.09
			10	—	**6.33**	3.8				10	91.90%	**15.8**	5.92
2	12	166	2	0.00%	11.52	600	7	14	192	2	8.03%	15.33	600
			3	0.00%	8.98	600				3	14.63%	13.44	600
			4	1.28%	8.21	600				4	30.82%	**11.61**	4.01
			5	13.62%	7.37	600				5	28.66%	**11.61**	5.52
			6	4.47%	**6.5**	3.17				6	87.48%	**11.61**	6.4
			7	24.83%	**6.5**	2.12				7	89.15%	**11.61**	7.81
			8	20.76%	**6.5**	2.3				8	91.68%	**11.61**	5.34
			9	18.56%	**6.5**	2.58				9	91.28%	**11.61**	5.8
			10	37.42%	**6.5**	2.72				10	92.86%	**11.61**	5.85
3	12	170	2	0.00%	12.83	600	8	14	210	2	3.96%	22.83	600
			3	10.91%	11.71	600				3	2.50%	15.47	600
			4	17.33%	**10.87**	2.86				4	2.18%	**15.47**	4.97
			5	13.71%	**10.87**	2.31				5	5.69%	**15.47**	4.9
			6	9.54%	**10.87**	2.39				6	0.00%	**15.47**	3.93
			7	0.00%	**10.87**	2.28				7	0.00%	**15.47**	3.35
			8	12.99%	**10.87**	2.52				8	90.65%	**15.47**	2.95
			9	0.00%	**10.87**	2.49				9	95.23%	**15.47**	6.01
			10	33.35%	**10.87**	2.71				10	93.09%	**15.47**	4.52
4	16	204	2	20.52%	27.88	600	9	14	178	2	4.08%	26.38	600
			3	46.34%	17.5	600				3	0.00%	**23.55**	3.16
			4	84.08%	**15.82**	10.48				4	0.00%	**23.55**	2.97
			5	—	**15.82**	264.33				5	0.00%	**23.55**	3.67
			6	—	**15.82**	13.22				6	0.00%	**23.55**	3.58
			7	—	**15.82**	17.92				7	0.00%	**23.55**	4.96
			8	—	**15.82**	16.58				8	88.91%	**23.55**	3.52
			9	—	**15.82**	14.92				9	94.49%	**23.55**	7.61
			10	—	**15.82**	13.12				10	94.75%	**23.55**	3.48
5	15	194	2	18.15%	17.44	600	10	19	206	2	—	17.59	600
			3	20.21%	14.27	600				3	—	12.84	600
			4	28.03%	11.34	600				4	—	10.04	600
			5	86.31%	10.15	600				5	—	**8.51**	43.96
			6	87.98%	9.26	600				6	—	**8.51**	3.25
			7	—	**8.82**	496.24				7	—	**8.51**	8.47
			8	—	**8.82**	34.83				8	—	**8.51**	6.71
			9	—	**8.82**	6.09				9	—	**8.51**	36.01
			10	—	**8.82**	5.92				10	—	**8.51**	21.74

Table 4: Results for large-size Southwestern K-ARCP instances

ins	$\lvert Q\rvert$	$\lvert B\rvert$	$\lvert K\rvert$	R-MIP	MIP-LS		ins	$\lvert Q\rvert$	$\lvert B\rvert$	$\lvert K\rvert$	R-MIP	MIP-LS	
				GAP (%)	Best OFV	Time (s)					GAP (%)	Best OFV	Time (s)
1	14	378	2	12.84%	2.11	1200	6	16	366	2	11.79%	3.21	1200
			3	23.39%	1.56	1200				3	0.00%	2.54	1200
			4	42.46%	1.27	1200				4	0.00%	**2.34**	115.43
			5	—	1.05	1200				5	12.76%	**2.34**	97.84
			6	—	0.94	1200				6	89.14%	**2.34**	511.33
			7	93.99%	0.8	1200				7	—	**2.34**	121.27
			8	—	0.78	1200				8	79.05%	**2.34**	113.13
			9	94.62%	**0.72**	76.64				9	86.35%	**2.34**	119.23
			10	—	**0.72**	75.61				10	80.71%	**2.34**	154.07
2	14	366	2	0.09%	2.26	1200	7	16	370	2	14.89%	2.03	1200
			3	13.61%	1.78	1200				3	24.98%	1.43	1200
			4	21.24%	1.23	1200				4	50.46%	1.06	1200
			5	40.07%	0.99	1200				5	87.25%	0.88	1200
			6	80.19%	**0.98**	28.25				6	—	**0.78**	18.51
			7	—	**0.98**	22.15				7	—	**0.78**	18.93
			8	61.23%	**0.98**	26.11				8	—	**0.78**	15.82
			9	70.32%	**0.98**	22.87				9	—	**0.78**	18.87
			10	70.13%	**0.98**	23.19				10	—	**0.78**	22.51
3	14	376	2	24.75%	2.34	1200	8	15	368	2	16.86%	2.85	1200
			3	41.50%	1.7	1200				3	11.66%	**2.49**	64.2
			4	55.96%	1.32	1200				4	23.91%	**2.49**	35.7
			5	53.65%	**1.13**	123.84				5	88.43%	**2.49**	32.17
			6	68.93%	**1.13**	71.05				6	91.33%	**2.49**	32.64
			7	—	**1.13**	57.67				7	92.62%	**2.49**	56.03
			8	—	**1.13**	39.59				8	79.48%	**2.49**	67.81
			9	—	**1.13**	39.85				9	92.33%	**2.49**	73.41
			10	—	**1.13**	64.53				10	79.87%	**2.49**	69.78
4	13	382	2	21.84%	1.6	1200	9	19	374	2	—	3.02	1200
			3	—	1.16	1200				3	—	2.39	1200
			4	42.21%	1.07	1200				4	—	1.84	1200
			5	53.43%	0.88	1200				5	—	1.81	1200
			6	59.99%	0.81	1200				6	—	1.67	1200
			7	92.93%	0.64	1200				7	—	1.41	1200
			8	62.07%	**0.59**	531.25				8	—	1.31	1200
			9	—	**0.59**	44.89				9	—	**1.22**	108.59
			10	—	**0.59**	87.14				10	—	**1.22**	123.89
5	15	374	2	11.24%	2.18	1200	10	15	368	2	18.67%	2.93	1200
			3	18.28%	1.68	1200				3	63.85%	2.18	1200
			4	37.74%	1.31	1200				4	87.54%	1.79	1200
			5	42.59%	1.11	1200				5	92.66%	1.57	1200
			6	45.28%	1.01	1200				6	95.50%	1.37	1200
			7	—	**0.94**	22.4				7	89.01%	1.2	1200
			8	91.80%	**0.94**	80.54				8	94.47%	**1.04**	121.56
			9	58.26%	**0.94**	28.91				9	—	**1.04**	119.73
			10	64.75%	**0.94**	29.38				10	—	**1.04**	117.38

As the number of vehicles increases MIP-RLS is able to verify the optimality of a feasible solution using Proposition 2 and corollary 1 in more cases. In general the optimality of more instances can be verified in Simplified Istanbul instance. However, when the number of vehicles increases in most of the instances in both Southwestern and Simplified networks the optimality is achieved. When the number of vehicles is less than 5, the optimality cannot be verified in more instances. However, in Section 5.1, we showed that in instances with less than 5 vehicles, if we can verify a lower bound the obtained feasible solution from the MIP-RLS gives very good optimality gaps.

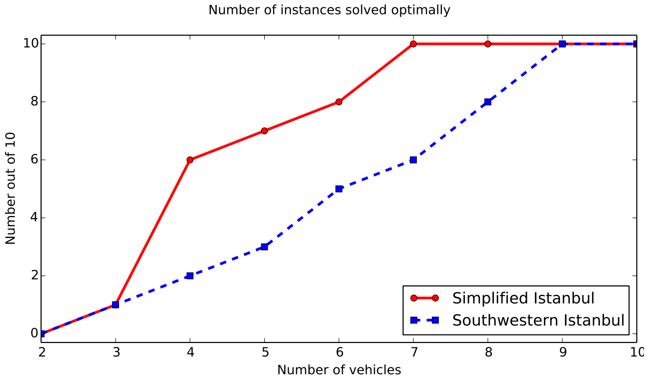


Figure 8: MIP-RLS optimality analysis

Figure 9 depicts a summary on the performance of the rich local search step of the MIP-RLS algorithm. Plot (a) shows the average improvement percentage by the RLS steps. Generally, when the number of road clearance teams increases, the improvement decreases. This is mainly because with more vehicles, ISBP finds the optimal solution of K-ARCP more frequently. For instance, in all the Simplified network instances with 9 and 10 vehicles, the optimal solution was found by ISBP. Moreover, we can see that on average the improvement percentage of the Simplified instances are higher than those of Southwestern network with the same number of vehicles. This is because the graph in the Simplified instances are smaller compared to Southwestern and more successful moves are applied even in the shorter time-limits (600 seconds for Simplified and 1200 seconds for Southwestern). The latter case can be approved in plot (b) as well. Plot (b) of Figure 9 shows the number of successful RLS moves for the tested instances. Again, in Simplified instances with 9 and 10 vehicles, no improvement move is applied since ISBP gives the optimal solution in all the tested instances. As the number of vehicles increases, this number decreases in general. The reason of this behavior is explained in the following; plot (c) in Figure 9 gives the information on when the improvement moves are done in average over all the tested instances. With each network (Simplified or Southwestern) and each number of vehicles ($k = 2, ..., 10$), the y-axis value of the circle shows the average time that an improvement move is found and applied and the size of the circle shows the frequency of such improvement moves. For instance, with 2 vehicles on the Simplified Istanbul network, the average time for an improvement move to be found and applied was 152.60 seconds and there has been 41 of such improvement moves. On the other hand, for instances with 3 vehicles on the Southwestern network, on the average 892.44 seconds is spent to find and apply an improvement move and 17 of such moves have been found in total. This plot shows that "in general" as the number of vehicles increases the average time required to find an improvement move decreases. However, we know that once the number of road clearance teams increases, the number of improving moves decreases from plot (b). This is because when the number of vehicles increases, the performance of ISBP becomes better and it finds optimal or near-optimal solutions that can turn into optimal solutions with small number of improving moves. This is why in plot (b) the number of successful moves decreases as k increases. It is also observed that when the larger network is considered, from Simplified to Southwestern, the required time to find an improving move is generally higher.

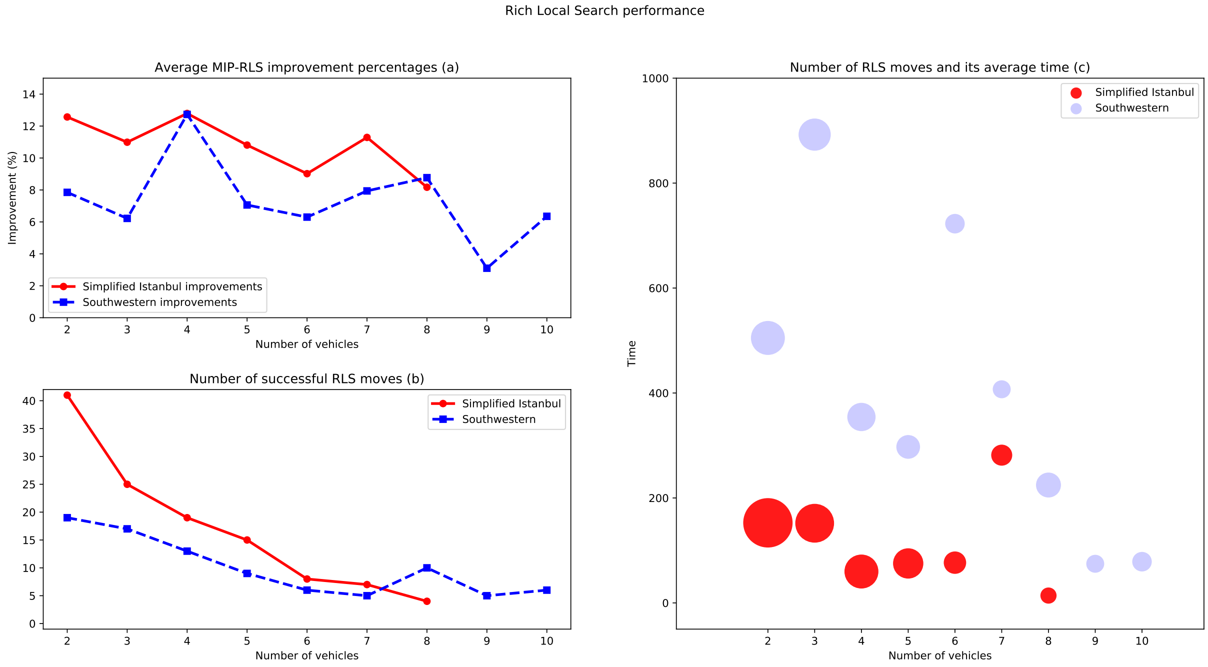


Figure 9: RLS performance analysis

Table 5 summarizes the results of the tested K-ARCP instances. We have set 2 to 10 vehicles in all the tested instances. Note that since timing conflicts do not occur in instances with single vehicle, they can be solved optimally in relatively short time intervals. For the Simplified network instances, the given time-limit was 600 seconds. The average relative mid-MIP gap of the R-MIP for the tested instances in the time-limit is reported in this table. This gap is extracted from the solver once the given time-limit is reached and the model has not been solved to optimality. In almost all the instances, the 600 time-limit is fully used. Without the optimal solution of R-MIP, the local search procedure developed in the literature cannot be initiated and these gaps are not the optimality gap to the K-ARCP. However, using the MIP-RLS we can find optimal or near-optimal solutions in all the tested instances. For example, in the Simplified instances with 10 vehicles, all the instances were solved on average within 6.97 seconds. For the Southwestern instances, the time-limit was set to be 1800 seconds and the summary of the results for those instances is also given in the right side of Table 5.

Table 5: Summary of the results on large-size K-ARCP instances

	Simplified Istanbul			Southwestern		
	R-MIP		MIP-RLS	R-MIP		MIP-RLS
K	Gap	Time	Time	Gap	Time	Time
2	9.74%	506.45	291.26	14.77%	1800	850.64
3	19.71%	526.17	219.13	24.66%	1682.93	1007.45
4	28.67%	563.35	115.52	40.17%	1733.61	617.34
5	29.62%	561.43	49.60	58.86%	1800	437.04
6	27.12%	578.26	37.86	75.77%	1800	590.59
7	19.30%	581.83	55.97	92.14%	1800	300.81
8	50.83%	581.82	8.42	78.02%	1800	145.48
9	50.00%	558.77	9.13	80.38%	1800	65.29
10	73.90%	600	6.97	73.87%	1800	76.74

6. Conclusions and remarks

In this paper, we have considered the coordinated multicrew arc routing for optimal connectivity in a post disaster condition. In this problem, the paths of K work crews in a post-disaster condition where some of the streets are blocked is optimized. These work crews are capable of reopening the blocked road segments by spending additional opening time on them. The debris accumulated on the roads after a disaster impedes accessibility to critical locations such as hospitals and fire stations. The paths of the work crews should be coordinated to prevent traversing blocked edges without opening them. We developed an efficient heuristic algorithm with a novel initial solution generation step. We call our algorithm the MIP-based heuristic with Rich Local Search (MIP-RLS). The initial solutions for the proposed MIP-RLS procedure are generated using an optimization model that is created by pre-processing the original problem. This model is a binary programming optimization model namely Initial Solution Binary Problem (ISBP). After the generation of the initial solutions, new and rich problem-specific local search moves are defined and implemented. Since the structure of problem is unique in terms of feasibility and optimality of the paths, we developed a new novel search procedures that can be used in problems with multiple work crews. We further analysed the performance of our algorithm and compared it with the optimal solution by a number of propositions. In this study, we tested our proposed algorithm on instances from Istanbul road network. While in the literature the largest tested instances consider up to 6 separated components and 4 work crews, we tested instances with up to 19 separated components, and up to 10 work crews. According to our numerical tests, our proposed MIP-RLS finds a feasible solution for all instances within less than 2 minutes.

In a more realistic setting, the structure of this study can be changed to cases where the road clearance crews may differ in terms of their capabilities since the machinery, equipment, tools, consumables and operating personnel required for a specific restoration task may show variability. Given a set of limited resources, team forming can be a part of the problem, after an initial assessment of the required tasks. Moreover, some equipment may even be repositioned among the teams working at different locations. The obtained results can be compared to perform a study on the use of these problems in post-disaster decision support. While we considered the problem of connecting all the components without prioritization, in disaster response phase it is crucial to reconnect the critical components first and then aim to reconnect all other components. In this case, we still need to be able to solve the second step of the problem that aims to reconnect the entire network in a short time, and still the validity and importance of the problem in this study hold.

References

Ahmadi, Morteza, Seifi, Abbas, & Tootooni, Behnam. 2015. A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transportation Research Part E: Logistics and Transportation Review*, **75**(Supplement C), 145 – 163.

Ajam, Meraj, Akbari, Vahid, & Salman, F. Sibel. 2019. Minimizing latency in post-disaster road clearance operations. *European Journal of Operational Research*, **277**(3), 1098 – 1112.

Akbari, Vahid, & Salman, F. Sibel. 2017a. Multi-vehicle prize collecting arc routing for connectivity problem. *Computers and Operations Research*, **82**, 52 – 68.

Akbari, Vahid, & Salman, F. Sibel. 2017b. Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. *European Journal of Operational Research*, **257**(2), 625 – 640.

Akbari, Vahid, & Shiri, Davood. 2021. Weighted online minimum latency problem with edge uncertainty. *European Journal of Operational Research*.

Anaya-Arenas, A. M., Renaud, J., & Ruiz, A. 2014. Relief distribution networks: a systematic review. *Annals of Operations Research*, **223**(1), 53–79.

Barbalho, Thiago Jobson, Santos, Andréa Cynthia, & Aloise, Dario José. 2021. Metaheuristics for the work-troops scheduling problem. *International Transactions in Operational Research*, **n/a**(n/a).

Briskorn, Dirk, Kimms, Alf, & Olschok, Denis. 2020. Simultaneous planning for disaster road clearance and distribution of relief goods: a basic model and an exact solution method. *OR Spectrum*, **42**(3), 591 – 619.

Cartes, Pablo, Echaveguren Navarro, Tomás, Giné, Alondra Chamorro, & Binet, Eduardo Allen. 2021. A cost-benefit approach to recover the performance of roads affected by natural disasters. *International Journal of Disaster Risk Reduction*, **53**, 102014.

Caunhye, Aakil M., Aydin, Nazli Yonca, & Duzgun, H. Sebnem. 2020. Robust post-disaster route restoration. *OR Spectrum*, **42**(4), 1055–1087.

Cavdaroglu, Burak, Hammel, Erik, Mitchell, John E, Sharkey, Thomas C, & Wallace, William A. 2013. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. *Annals of Operations Research*, 1–16.

Çelik, Melih. 2016. Network restoration and recovery in humanitarian operations: Framework, literature review, and research directions. *Surveys in Operations Research and Management Science*, **21**(2), 47 – 61.

Çelik, Melih, Ergun, Özlem, & Keskinocak, Pınar. 2015. The Post-Disaster Debris Clearance Problem Under Incomplete Information. *Operations Research*, **63**(1), 65–85.

Coco, Amadeu A., Duhamel, Christophe, & Santos, Andréa Cynthia. 2020. Modeling and solving the multi-period disruptions scheduling problem on urban networks. *Annals of Operations Research*, **285**(1), 427–443.

Duque, P. M., & Sörensen, K. 2011. A GRASP metaheuristic to improve accessibility after a disaster. *OR Spectrum*, **33**, 525–542.

Duque, Pablo A Maya, Dolinskaya, Irina S, & Sörensen, Kenneth. 2016. Network repair crew scheduling and routing for emergency relief distribution problem. *European Journal of Operational Research*, **248**(1), 272–285.

Duran, Serhan, Gutierrez, Marco A., & Keskinocak, Pinar. 2011. Pre-Positioning of Emergency Items for CARE International. *Interfaces*, **41**(3), 223–237.

Golla, Arpan Paul Singh, Bhattacharya, Shankha Pratim, & Gupta, Sumana. 2020. The accessibility of urban neighborhoods when buildings collapse due to an earthquake. *Transportation Research Part D: Transport and Environment*, **86**, 102439.

Kasaei, Maziar, & Salman, F. Sibel. 2016. Arc routing problems to restore connectivity of a road network. *Transportation Research Part E: Logistics and Transportation Review*, **95**, 177–206.

Kim, Karl, Renne, John, & Wolshon, Brian. 2019. Minding the gap: New section in transportation research, Part D to focus on disasters and resilience. *Transportation Research Part D: Transport and Environment*, **77**, 335 – 336.

Li, Shuanglin, & Teo, Kok Lay. 2019. Post-disaster multi-period road network repair: work scheduling and relief logistics optimization. *Annals of Operations Research*, **283**(1), 1345–1385.

Li, Shuanglin, Ma, Zujun, & Teo, Kok Lay. 2020. A new model for road network repair after natural disasters: Integrating logistics support scheduling with repair crew scheduling and routing activities. *Computers & Industrial Engineering*, **145**, 106506.

Mete, H. O., & Zabinsky, Z. B. 2010. Stochastic optimization of medical supply location and distribution in disaster management. *International Journal of Production Economics*, **126**(1), 76 – 84.

Minas, J.P., Simpson, N.C., & Tacheva, Z.Y. 2020. Modeling emergency response operations: A theory building survey. *Computers & Operations Research*, **119**, 104921.

Moreno, Alfredo, Munari, Pedro, & Alem, Douglas. 2019. A branch-and-Benders-cut algorithm for the Crew Scheduling and Routing Problem in road restoration. *European Journal of Operational Research*, **275**(1), 16–34.

Moreno, Alfredo, Munari, Pedro, & Alem, Douglas. 2020a. Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. *Computers & Operations Research*, **119**, 104935.

Moreno, Alfredo, Alem, Douglas, Gendreau, Michel, & Munari, Pedro. 2020b. The heterogeneous multicrew scheduling and routing problem in road restoration. *Transportation Research Part B: Methodological*, **141**, 24–58.

Morshedlou, Nazanin, González, Andrés D., & Barker, Kash. 2018. Work crew routing problem for infrastructure network restoration. *Transportation Research Part B: Methodological*, **118**(dec), 66–89.

Oruc, Buse Eylul, & Kara, Bahar Yetis. 2018. Post-disaster assessment routing problem. *Transportation Research Part B: Methodological*, **116**, 76 – 102.

Özdamar, Linet, & Ertem, Mustafa Alp. 2015. Models, solutions and enabling technologies in humanitarian logistics. *European Journal of Operational Research*, **244**(1), 55 – 65.

Renne, John, Wolshon, Brian, Murray-Tuite, Pamela, & Pande, Anurag. 2020. Emergence of resilience as a framework for state Departments of Transportation (DOTs) in the United States. *Transportation Research Part D: Transport and Environment*, **82**, 102178.

Sanci, Ece, & Daskin, Mark S. 2019. Integrating location and network restoration decisions in relief networks under uncertainty. *European Journal of Operational Research*, **279**(2), 335 – 350.

Sanci, Ece, & Daskin, Mark S. 2021. An integer L-shaped algorithm for the integrated location and network restoration problem in disaster relief. *Transportation Research Part B: Methodological*, **145**, 152–184.

Sayarshad, Hamid R., Du, Xinpi, & Gao, H. Oliver. 2020. Dynamic post-disaster debris clearance problem with repositioning of clearance equipment items under partially observable information. *Transportation Research Part B: Methodological*, **138**, 352 – 372.

Shin, Youngchul, Kim, Sungwoo, & Moon, Ilkyeong. 2019. Integrated optimal scheduling of repair crew and relief vehicle after disaster. *Computers & Operations Research*, **105**, 237 – 247.

Shiri, Davood, Akbari, Vahid, & Salman, F. Sibel. 2020. Online routing and scheduling of search-and-rescue teams. *OR Spectrum*, **42**(3), 755–784.

Sokat, Kezban Yagci, Dolinskaya, Irina S., Smilowitz, Karen, & Bank, Ryan. 2018. Incomplete information imputation in limited data environments with application to disaster response. *European Journal of Operational Research*, **269**(2), 466 – 485.

Sun, Jingran, & Zhang, Zhanmin. 2020. A post-disaster resource allocation framework for improving resilience of interdependent infrastructure networks. *Transportation Research Part D: Transport and Environment*, **85**, 102455.

Tuzun Aksu, Dilek, & Ozdamar, Linet. 2014. A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation. *Transportation Research Part E: Logistics and Transportation Review*, **61**, 56–67.

Vodák, Rostislav, Bíl, Michal, & Křivánková, Zuzana. 2018. A modified ant colony optimization algorithm to increase the speed of the road network recovery process after disasters. *International Journal of Disaster Risk Reduction*, **31**, 1092–1106.

Xu, Bei, & Song, Yuanbin. 2015. An Ant Colony-based Heuristic Algorithm for Joint Scheduling of Post-earthquake Road Repair and Relief Distribution. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, **13**(2), 632–643.

Zhou, Yaoming, Wang, Junwei, & Sheu, Jiuh-Biing. 2019. On connectivity of post-earthquake road networks. *Transportation Research Part E: Logistics and Transportation Review*, **123**, 1 – 16.