Online Routing and Scheduling of Search and Rescue Teams

Davood Shiri[a], Vahid Akbari[b], F. Sibel Salman[a,*]

[a]*College of Engineering, Koç University, Sariyer, Istanbul 34450, Turkey*
[b]*Nottingham University Business School, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom*

Abstract

We study how to allocate and route search-and-rescue (SAR) teams to areas with trapped victims in a coordinated manner after a disaster. We propose two online strategies for these time-critical decisions considering the uncertainty about the operation times required to rescue the victims and the condition of the roads that may delay the operations. First, we follow the theoretical competitive analysis approach that takes a worst-case perspective and prove lower bounds on the competitive ratio of the two variants of the defined online problem with makespan and weighted latency objectives. Then, we test the proposed online strategies and observe their good performance against the offline optimal solutions on randomly generated instances.
Keywords: Disaster logistics, search-and-rescue, online optimization, makespan, latency, multiple teams, edge blockage

1. Introduction

Search-and-rescue (SAR) in a disaster situation, either natural or man-made, involves time-critical activities to locate a large number of victims, as well as saving them from entrapment. SAR operations go in stages. In the assessment stage facts are gathered and target areas are determined. Once areas where people are entrapped have been identified by means of various technology (see Ganz et al. (2015)), search operations involving multiple teams should be put into action in a coordinated way, following a well-designed action plan that should be prepared as soon as possible. Next, the teams are dispatched to reach the victims under dire conditions such as blocked roads. Via systematic search techniques and rescue operations, trapped victims are secured and provided medical aid as necessary. SAR teams not only include trained personnel with specialized skills and equipment, but also other supporting personnel such as paramedics, policemen and firemen. Furthermore, the teams have the ability to communicate with each other, e.g. through a mobile ad hoc network (Anjum et al., 2015).

Among the factors that prolong SAR operations, the mobilization and dispatching of the rescue teams play an important role (Statheropoulos et al., 2015). As stated in Poteyeva et al. (2007), according to Wenger (1990) and the literature therein, there is consensus that in general SAR operations are hampered by problems of timely access. In fact, it is claimed that 80% to 90% of entrapped victims who survive are

*Corresponding author
Email addresses: dshiri14@ku.edu.tr (Davood Shiri), vahid.akbari@nottingham.ac.uk (Vahid Akbari), ssalman@ku.edu.tr (F. Sibel Salman)

recovered in the first 48 hours after the disaster impact, and that many more entrapped victims could survive with timely delivery of appropriate medical care (Poteyeva et al., 2007).

Research on disaster preparedness and response has focused mostly on pre-positioning of emergency supplies, emergency response facility location, design of relief supply networks, distribution of relief aid, emergency transportation, resource allocation and evacuation (Tang et al., 2018), and neglected optimization of SAR operations largely. In this paper, we optimize dispatching of heterogeneous SAR teams differing in their SAR operation capacity, that involves both their allocation to areas with trapped victims and designing their routes to reach the victim locations. Sheu (2007) cited handling the operational uncertainties as one of the most important challenges of emergency logistics management. We propose online strategies for the SAR teams considering the uncertainty about the operation times required to rescue the victims and the condition of the roads. We define an online optimization problem on a network where the locations with trapped victims are nodes to be reached and processed, and the road connections between them are edges in this network, which might be blocked. We consider two versions of the problem with two objectives, namely, minimizing the completion time of the SAR operations (makespan) and minimizing the total weighted latency until victims are rescued, where latency of each victim location (node) is weighted by the percentage of victims trapped there. The second objective aims to minimize the overall waiting time of the people until they get rescued.

In theoretical analysis of online strategies, the quality of the strategies is evaluated by the *competitive analysis* approach. That is, the performance of the strategy that operates under incomplete information, i.e. the *online* strategy, is compared with the performance of the optimal strategy that operates in presence of complete information, i.e. the *offline* strategy. This type of analysis was first suggested in (Sleator & Tarjan, 1985) and later called *competitive analysis* in (Karlin et al., 1988). To evaluate the performance of online strategies, the notion of *competitive ratio* has been introduced by Sleator & Tarjan (1985) and adopted by many researchers. For a *deterministic* online strategy, the competitive ratio is the maximum ratio of the cost of the online strategy to the cost of the offline strategy over all instances of the problem. For a *randomized* online strategy, the expected competitive ratio is the maximum ratio of the expected cost of the online strategy to the cost of the offline strategy over all instances of the problem. For our two online problems with makespan and weighted latency objectives, we analyze how small competitive ratio a deterministic online strategy may achieve. By finding lower bounds on the competitive ratio for the two problems, we show that any deterministic online strategy cannot get closer to the offline optimum than the corresponding lower bound. This result shows the value of having complete information at the moment the problem is solved.

Our contributions can be summarized as follows. We analyze an online optimization approach for the search and rescue problem with multiple teams for the first time. With the aim of optimizing the SAR operations in a mass casualty incident, we introduce a new problem to the online optimization literature. We derive theoretical lower bounds on the competitive ratio of the two variants of the proposed problem with different objectives. We introduce two alternative novel online strategies that are shown to run in very short time on realistic-sized instances.

The remainder of the paper is organized as follows. In the next section, we provide a review of relevant research. In Section 3, we introduce two online optimization problems. We provide mixed integer programming (MIP) models for the offline versions of these problems in Section 4. In Section 5, we derive lower bounds on the competitive ratio of deterministic strategies for our proposed online problems. We present two deterministic online strategies for our problems in Section 6. We provide our computational study in Section 7. Finally, we conclude in Section 8.

2. Literature review

Since our study focuses on online optimization of routing and allocation decisions for rescue units in disaster response, our primary focus in this section will be on studies in three areas. First, we discuss articles on online routing problems with edge blockages. Next, we review the work related to the scheduling of rescue units in the disaster response phase. Finally, we give an overview of articles on disaster relief routing.

2.1. Routing problems with online blocked edges

Analyzing routing problems with online blocked edges in networks has a rich background. These problems involve one or more traveling agents on a road network modelled by a graph. An online blocked edge is not known to the traveling agents initially and is revealed whenever one of its end-nodes is visited by at least one of the agents. Typically, it is assumed that there exists a known number of blocked edges but their locations are not known. Our problem also tackles edge blockages but in addition considers uncertainty in search times as well.

2.1.1. Single-agent problems

Using online blocked edges to model traffic uncertainties was introduced for the Canadian Traveler Problem (CTP) in Papadimitriou & Yannakakis (1991). The CTP is essentially a variation of the shortest path problem on graphs that are partially observable and contain online blocked edges. It is proven that devising an online strategy with a bounded competitive ratio is PSPACE-complete for the CTP (Papadimitriou & Yannakakis (1991)). Several variations of the CTP are considered in Bar-Noy & Schieber (1991). The k-CTP, the version of CTP where an upper bound k on the number of blocked edges is given as input, is defined in Bar-Noy & Schieber (1991) for the sake of computational ease. Furthermore, it is shown that for arbitrary k, the problem of designing an online strategy that guarantees the minimum travel cost is PSPACE-complete.

Traffic uncertainties have been taken into consideration for the Traveling Salesman Problem (TSP) as well, and modelled with online blocked edges. A variation of the TSP that involves finding a shortest tour including all nodes on a complete edge-weighted graph where there are k online blocked edges in the graph, is investigated in Liao & Huang (2014). Since the problem has the same type of uncertainty as of the uncertainty in the CTP, this problem is called the online Covering Canadian Traveler Problem (CCTP). The online CCTP is analyzed from the (worst-case) competitive ratio perspective. An efficient touring strategy is given within an $o(\sqrt{k})-$competitive ratio.

Another related problem is the online Steiner TSP (STSP) in which a salesman is required to visit only a certain subset of the vertices in presence of k online blocked edges. This problem is investigated in Zhang et al. (2015) where a tight lower bound on the competitive ratio of deterministic online strategies is proven together with an exponential-time optimal deterministic online strategy. A polynomial-time asymptotically optimal deterministic online strategy is also proposed. The performance of this polynomial-time online strategy is tested on sparse randomly generated networks with k randomly generated online blocked edges in Zhang et al. (2015). The STSP with k online advanced blocked edges, where the salesman receives blockage messages when he is at a certain distance to the respective blocked edges is studied in Zhang et al. (2016). Such road blockages are referred to as advanced blocked edges. In that study a lower bound on the competitive ratio of deterministic online strategies is proven. In addition, a polynomial-time deterministic online strategy is proposed and tested on sparse randomly generated networks.

In addition to the above discussed problems, the minimum latency problem with k online blocked edges has been recently investigated in Zhang et al. (2019), where a lower bound on the competitive ratio of deterministic online strategies is proven. Moreover, two efficient deterministic online strategies are proposed and evaluated on randomly generated networks.

Here we emphasize that our problem is a multi-agent problem with multiple origin and destination nodes. We conduct a competitive ratio analysis and in addition, investigate the performance of our proposed strategies by computational tests.

2.1.2. Multi-agent problems

A generalization of the k-CTP with multiple agents, which is called the online multi-agent k-CTP, has also been investigated in the literature. The online multi-agent k-CTP with different levels of communication between agents is studied in Zhang et al. (2013). Lower bounds on the competitive ratio of deterministic online strategies are proven. Also, two deterministic online strategies for special graphs which contain several O-D edge-disjoint paths are proposed. In Shiri & Salman (2017), improved lower bounds on the competitive ratio of deterministic online strategies for the online multi-agent k-CTP are given. An optimal deterministic online strategy for O-D edge-disjoint graphs is provided as well. Randomized online strategies for the online multi-agent k-CTP with different levels of communication between agents are proposed in Shiri & Salman (2019a). Lower bounds on the expected competitive ratio of randomized online strategies are provided together with an optimal randomized strategy for O-D edge-disjoint graphs. The multi-agent k-CTP is investigated from a computational point of view in Shiri & Salman (2019b). An efficient heuristic deterministic online strategy is proposed and tested on real city road networks as well as randomly generated networks.

We would like to stress that in all of the articles which investigate the online multi-agent k-CTP, competitive ratios of deterministic (or randomized) online strategies have been only proven for O-D edge-disjoint graphs or graphs which contain a high number of O-D edge-disjoint paths. To the best of our knowledge, the online multi-agent k-CTP is the only multi-agent problem in the literature which uses online blocked edges to model traffic uncertainties. In none of the related studies discussed here, a multi-origin, multi-destination and multi-agent problem has been analyzed.

2.2. Scheduling rescue units in disaster response

In disaster operations management, a challenging task for rescue organizations is to assign and schedule the rescue units to emergency incidents under time pressure in order to reduce the overall resulting harm. The scheduling of collaborative rescue units is investigated in Rolland et al. (2010), where a resource-constrained project scheduling problem is proposed and two meta-heuristics are presented for its solution. A quadratic binary programming model and a heuristic are developed for scheduling collaborative rescue units in Wex et al. (2013). Scheduling and allocation of rescue units to incidents are addressed in Wex et al. (2014) and a mixed integer non-linear programming model is developed. A rather sophisticated heuristic is developed in Schryen et al. (2015) for the joint allocation of rescue units and the scheduling of incidents under different conditions of collaboration, based on scheduling theory. The proposed heuristic is benchmarked against a heuristic best-practice behavior and against lower bounds from a quadratic programming relaxation. a MIP model is developed in Bodaghi & Ekambaram (2016) to minimize the relief operation completion times required for all incidents by optimally assigning and scheduling various teams. For a small case study with four rescue units, an optimal solution is obtained using a commercial solver. Although the model could account for it, the case study does not involve multi-capability rescue units. In Rauchecker & Schryen (2019), scheduling rescue units during disaster response is optimized such that each rescue unit may offer different capabilities and each incident may require multiple capabilities. Weights are assigned to an incident according to its severity level and the total weighted completion time is minimized. For the related binary linear programming problem, a branch-and-price algorithm which can serve as both an exact algorithm and a heuristic solution procedure under limited time is developed.

Departing from the above work, dynamic nature of the post-disaster conditions is modeled in Chen & Miller-Hooks (2012) for the problem of deploying SAR teams to disaster sites. In this problem, assistance requirements arrive dynamically, and the objective is to maximize the total expected number of people that can be saved. Decisions are taken dynamically as more information is obtained while survival likelihood of people decreases by time. A multi-stage stochastic program is developed and solved by means of solving a series of interrelated two-stage stochastic programs with recourse.

Previous work on this topic does not address the routing of the rescue units on a graph so far, whereas we address routing decisions in addition to allocating the SAR teams to the rescue tasks. Furthermore, in none of the above studies availability of incomplete information has been considered, while our current study aims to find online strategies under uncertainty of connections between the locations and rescue times for the tasks.

2.3. Routing in disaster response

Relief routing literature mainly focuses on relief item distribution, which aims to find an efficient, effective and equitable distribution of pre-positioned relief items to people in need. Several studies approached optimization problems on this topic via deterministic models. In Yan & Shih (2009), a multi-objective and multi-commodity network flow problem is defined for joint roadway repair and relief distribution, with the objective of minimizing the total completion time of these activities. Minimization of total delivery time or the latest arrival time of a vehicle is studied under a deterministic setting in Campbell et al. (2008) and Ozkapici et al. (2016). A fuzzy multi-objective linear programming model is proposed in Tzeng et al. (2007) for designing relief delivery systems. The objective is to minimize total relief costs of primary and secondary disasters. Another routing problem that involves routing teams is aimed at assessment of needs and damage following a disaster (Huang et al. (2013)). The objective is to minimize the sum of arrival times to beneficiaries, that is, total latency. Last mile distribution with multiple deliveries under multiple criteria is addressed in Ferrer et al. (2018), where a model is developed to find the optimal routes for transporting aid from the supply nodes to the demand nodes, as well as the required vehicle fleet size.

Models addressing uncertainty in relief aid distribution are discussed next. In Hoyos et al. (2015), a review of models with stochastic components in disaster operations management is provided and several articles on relief distribution are discussed. In addition to the relevant studies that can be found therein, recent studies on the subject include Alem et al. (2016), Zhang et al. (2012), Elci et al. (2018), Wang & Nie (2019) and Hu et al. (2019). In Alem et al. (2016), distribution of multiple types of relief aid commodities via heterogeneous vehicles over multiple time periods and multiple scenarios is optimized. In the proposed two-stage model, uncertainty of supply and demand amounts, and functionality of edges (edge blockage) are represented by discrete scenarios. The first stage decisions are pre-positioning of supplies, while second stage decisions involve distribution (without forming tours encompassing demand points). A fix and solve type of heuristic is provided. A local search heuristic together with an exact model for the multiple-disaster multiple-response emergency team allocation problem is studied in Zhang et al. (2012), while considering the stochastic occurrence of a secondary disaster. Addressing post-disaster uncertainty in demands and transportation network conditions, a stochastic last mile relief network design problem is defined and a stochastic optimization model that addresses accessibility and equity is developed in Elci et al. (2018). In Wang & Nie (2019), traffic congestion effects are incorporated into the problem that addresses relief aid prepositioning and its post-disaster transportation. A mixed integer nonlinear programming model is developed and the generalized Benders decomposition algorithm is employed for its solution. A multi-stage stochastic programming model for disaster relief distribution with consideration of uncertain and dynamic road capacity is solved in Hu et al. (2019). The authors propose a solution methodology based on the progressive hedging algorithm. Uncertainty of demand, incoming supplies, and route availability are addressed via a set of discrete scenarios and they are incorporated into a two-stage stochastic program that optimizes location, transportation and fleet sizing decisions in Moreno et al. (2018). Three heuristics are developed for the solution of the model.

Another approach used to address routing and distribution in disaster response under uncertainty is robust optimization. A robust bi-level optimization model is developed in Safaei et al. (2018) for a relief network design problem under uncertainty of demand and supply parameters. The Capacitated Vehicle Routing Problem and the Split Delivery Vehicle Routing Problem with uncertain travel times and demands are studied in Li & Chung (2019) for planning vehicle routes with the purpose of delivering critical supplies from a robust optimization perspective. Five objective functions are considered. A rolling horizon-based prediction and optimization framework, based on the so-called robust model predictive control approach, is proposed in Lu et al. (2016). The goal is to obtain robust relief distribution plans and adjust them in accordance with updated real-time information.

Several articles consider post-disaster relief routing, alone or mostly together with other decisions such as pre-disaster facility location or supply chain design, under network link disruptions that cause inaccessibility or increased travel times (Bruni et al. (2018), Diabat et al. (2019), Ahmadi et al. (2015), Aslan & Celik

(2019)). These studies are closer to ours compared to the above ones in terms of the type of uncertainty. A routing problem with multiple vehicles traveling in a damaged transportation network, and limited knowledge on the road travel times is defined and a heuristic algorithm is developed in Bruni et al. (2018). In Diabat et al. (2019), supply chain network design is addressed, where facilities and routes between them are subject to disruptions and the network might become inaccessible after a disaster. A bi-objective robust optimization model is provided to hedge against disruptions and a solution approach via Lagrangean relaxation is developed. A multi-depot location-routing problem is studied in Ahmadi et al. (2015), considering road closure within a two-stage stochastic programming framework. The design of a humanitarian relief supply network, subject to uncertainties in relief item demand and vulnerability of roads and facilities in the post-disaster stage is provided in Aslan & Celik (2019). A two-stage stochastic program is formulated and sample average approximation is used in its solution.

Differing from the above discussed studies, we analyze a routing problem from on an online optimization perspective. Hence, our methodology differs significantly from the work existing in the literature.

3. Problem description

We consider a mass casualty incident, such as an earthquake, flooding, wildfire, etc., in which several people have been trapped throughout a set of locations, which we call *critical* locations. It is not known a priori how many people are trapped in each critical location and how much time is required to rescue the people there. A mission is launched to rescue the victims throughout the area in shortest time. In a disaster situation, the urgent SAR operations are hindered by inaccessible roads as some road segments may easily be damaged or blocked, and rendered non-traversable. If information about which roads are blocked and how much time it takes to conduct a SAR operation at each critical location is available, the routes of the SAR teams can be planned at the beginning of the operation and executed according to this static plan. However, gathering the aforementioned information may take considerable time and rather than waiting for complete information, dispatching the teams under incomplete information, and adjusting their routes as information is revealed over the course of the operations saves time. Developing a routing and SAR strategy in such a case falls into the realm of online problems. In online problems information is revealed incrementally, while taking actions, and decisions must be made before all information is available. In such problems, the goal is to come up with a strategy that performs well over all possible data instances, compared to the optimal solution found assuming that all of the data is available a priori. The latter is called the offline optimal.

We next define an online optimization problem for the SAR setting described above. In this problem, we represent the road network in the disaster struck area by an undirected graph $G = (V, E)$, where V is the node set and E is the edge set. In an emergency situation, roads can be used in both directions so that the traveling time for each direction is symmetric. Also, this assumption is standard as most online routing problems in the literature are defined on undirected graphs (e.g., see (Zhang et al., 2019), (Zhang et al., 2015), (Shiri & Salman, 2019b), and (Shiri & Salman, 2017)). Let $S = \{v_1, v_2, ..., v_n\}$ ($S \subset V$) denote the set of critical nodes. The non-critical nodes are not only depots, but they may be junction points that can be used as intermediate nodes as well. For example, two end-nodes of a blocked edge can be two non-critical nodes. We note that in our problem, the criterion is to reach the victims in minimum time (rather than minimizing the time span of each tour). Therefore, the return times of the teams to the depots are irrelevant in this case and can be omitted in computing the objectives.

There are L heterogeneous SAR teams $T_1, T_2, ..., T_L$ in the graph who can communicate, meaning that once one of the unknown parameters is known to one of the teams, the information can be transferred to other SAR teams immediately. Team T_l ($l \in \{1, 2, ..., L\}$) is at a given (depot) node d_l in $V \setminus S$ at time zero and has a SAR operation rate r_l, where r_l denotes the amount of SAR operation that the team can perform in one unit of time. We assume that at most one SAR team can be assigned to a critical node. Given that the teams have different capacities, if a node requires more search time, a team with a higher capacity should be assigned to it.

An edge $e = (i, j) \in E$ is associated with a non-negative traveling time t_{ij}. There are k blocked edges in the graph, but these edges are not known to the teams at time zero. A blocked edge is learned online when at least one of the teams arrives at one of its end-nodes. This information is immediately communicated to the other teams. As a result, the routes of the teams may change as blockage information is obtained.

Let N_s be a positive integer number which represents the number of victims at critical node v_s ($s \in \{1, 2, ..., n\}$) and h_s be the required amount of SAR operation at this node to rescue all of the victims there. Both of these numbers are not known to the teams at time zero and their values are revealed online to the teams when exactly one of the teams performs the required operation. We assume that all of the victims at a critical node will be found and rescued when the required amount of SAR operation is performed at the node. The offline problem is different from the well-known m-TSP (Bektas, 2006) since we consider search times at the nodes and some nodes can be bypassed without searching them.

We investigate this problem with two different objectives. The first objective is to devise an online strategy such that the teams find and rescue all of the victims in minimum time (makespan minimization). We call the online problem under the makespan minimization objective P_M. For a critical node v_s ($s \in \{1, 2, ..., n\}$), let $latency_s$ represent the time taken from time zero until the completion of the h_s amount of SAR operation at v_s. In P_M, the objective is to devise an online strategy such that the teams rescue all of the victims in minimum time. That is, $\max_{v_s \in S} latency_s$ is minimized. The second objective is to design an online strategy such that the teams find and rescue all of the victims and the total weighted latency of the critical nodes containing victims is minimized. The weight of a critical node is taken as the ratio of the number of victims in the critical node to the total number of victims in the graph. We call the online problem under this objective P_{WL}. In P_{WL}, $\sum_{s=1}^{n} \frac{N_s}{\sum_{s=1}^{n} N_s} latency_s$ is minimized. We list the notation of the parameters in Table 1.

Remark 3.1. *The main focus of the problem is to handle the online uncertainties in a critical disaster situation. For an online optimization problem, offline solution approaches cannot be applied. This is because, once a piece of information is revealed, we cannot reset the problem to its initial setting. Instead, we should use the revealed information to make a decision from that point onward. We believe that the online solution approach is applicable in disaster response, where information uncertainty is prevalent and probabilistic knowledge about the uncertain information is hardly available. In this paper, we define an online optimization problem to study how to allocate and route SAR teams to areas with trapped victims in a coordinated manner after a disaster. We obtained expert opinion emphasizing the uncertainties about the operation times required* *to rescue the victims and the condition of the roads that may delay the operations. We propose online solution strategies for two variants of the defined online problem with makespan and weighted latency objectives.*

Table 1: Table of notations for the input parameters

Notation	**Description**
$G = (V, E)$	undirected graph G, node set V, edge set E
$S = \{v_1, v_2, ..., v_n\} \subset V$	set of critical nodes
h_s $(s \in \{1, 2, ..., n\})$	required amount of rescue effort at critical node v_s **(unknown)**
N_s $(s \in \{1, 2, ..., n\})$	number of victims at critical node v_s **(unknown)**
t_{ij}	traveling time of edge $e = (i, j) \in E$
$B = \{e_1, e_2, ..., e_k\}$	set of blocked edges **(unknown)**
$I = E - B$	set of intact edges
T_l $(l \in \{1, 2, ..., L\})$	SAR team l
r_l $(l \in \{1, 2, ..., L\})$	SAR operation rate of team l, i.e. effort/time
$D = \{d_1, d_2, ..., d_z\} \subset V - S$ $(z \leq m)$	set of depot nodes
$\Lambda = V - S - D$	set of intermediate nodes

4. The offline problem

In the offline problem, all input data, including the set of blocked edges, the number of people at each critical node and the amount of SAR operation required at each critical node, are known from the beginning. The problem is to find the routes of the SAR teams and determine when the victims at each critical node are rescued (i.e. their latency). The route of a SAR team starts from its initial node (depot), includes one or more critical nodes, as well as possibly other intermediate non-critical nodes between the critical nodes, and ends at a critical node. The offline problem with either of the objectives, namely the offline versions of P_M and P_{WL}, are both NP-hard. The minimum makespan problem is NP-hard because its single team special case generalizes the Steiner Traveling Salesman Problem (Cornuéjols et al. (1985), Letchford et al. (2013)). The total weighted latency problem is also NP-hard since the special case with a single team and uniform weights is the minimum latency problem, which is also called as the Traveling Repairman Problem (Bulhoes et al. (2018)). Therefore, for the solution of these problems, we resort to their MIP formulations.

We note that in the offline problem the set of blocked edges $B \subset E$ are known and it is assumed that the graph formed after the incident by the set of nodes V and the intact edges $I = E \setminus B$ is still connected. We show this graph by $G' = (V, I)$.

4.1. MIP models for makespan and weighted latency minimization problems

In this section, we formulate MIP models for the offline problems with makespan and weighted latency objective functions. In order to formulate the MIP models for these two problems, we adapt the approach proposed in Angel-Bello et al. (2017), which was shown computationally to lead to a strong formulation for the minimum latency problem with a single vehicle. Our problems differ in several aspects. Since the problem in Angel-Bello et al. (2017) involves only a single vehicle, it does not consider the coordination of the teams. It also does not involve search and rescue operations, and hence does not consider the edge blockage situation.

To formulate the problems, we define a multi-level network with $N+1$ levels, where $N = |S| - L + 1$. The value of N equals the maximum number of critical nodes that can be assigned to a single SAR team. In this multi-level network, level $N+1$ has L nodes including the depot of each SAR team. We note that we assume multiple teams might be pre-positioned in the same depot but we create a node for the depot of each of the SAR teams separately. Each of the levels from 2 to N has $L + |S|$ nodes, including the depots from level $N+1$ and all the critical nodes. Finally, level 1 has $|S|$ nodes including all the critical nodes. An illustration of this network is shown in Figure 1.

The first set of edges in this graph include the links between each of the depots from level r to each of the critical nodes in level $r-1$ $(r = 2, ..., N+1)$. Since in the transformed multi-level network, the depots and the critical nodes are copied in the levels, there is a direct link between each pair of nodes and the critical nodes are only visited to be served and not as intermediate nodes. As a result, the optimal solution to this problem gives disjoint paths. Moreover, we assume that the travel time between depot d_l and the critical node $s \in \{1, ..., |S|\}$ is the time required to go from node d_l to s plus the required service time from SAR team l in node s to finish the SAR operations. As a result, if we denote these costs by $\pi^l_{d_l s}$ $\forall l \in \{1, ..., L\}$ and $s \in \{1, ..., |S|\}$, we can see that $\pi^l_{d_l s} = c_{d_l s} + \frac{h_s}{r_l}$, where c_{ij} shows the shortest path distance from node i to node j in graph G' and $\frac{h_s}{r_l}$ gives the required time for team l to fulfill the SAR operations at node s.

The second set of edges in this graph are the links between the critical nodes from level r to critical nodes in level $r-1$, $(r = 2, ..., N)$. Similar to above, since the optimal paths are disjoint, we can say that once a team visits a critical node, it fulfills its SAR operations and then leaves it. Moreover, for each pair of critical nodes, there are L paths with corresponding costs for each of the teams. As a result, we define parameters $\pi^l_{ss'}$ $\forall l \in \{1, ..., L\}$ and $s, s' \in \{1, ..., |S|\}, s \neq s'$, where $\pi^l_{ss'}$ shows the travel time from critical node s to critical node s' plus the required SAR operation time for team l at critical node s' ($\pi^l_{ss'} = c_{ss'} + \frac{h_{s'}}{r_l}$). In Figure 1, only one of such edges between each pair of critical nodes is shown.

Using the multi-level network that we described above and illustrated in Figure 1, we generate MIP models for the offline problems. In the following, first we give the model for the makespan minimization problem, P_M and then present the model developed for the latency minimization problem, P_{WL}.

4.1.1. MIP model of the offline makespan minimization problem

In the makespan minimization problem, the objective is to minimize the time at which all the critical nodes are served. This corresponds to the time of serving the last critical node. In the following we first give the model and then describe the decision variables and constraints.

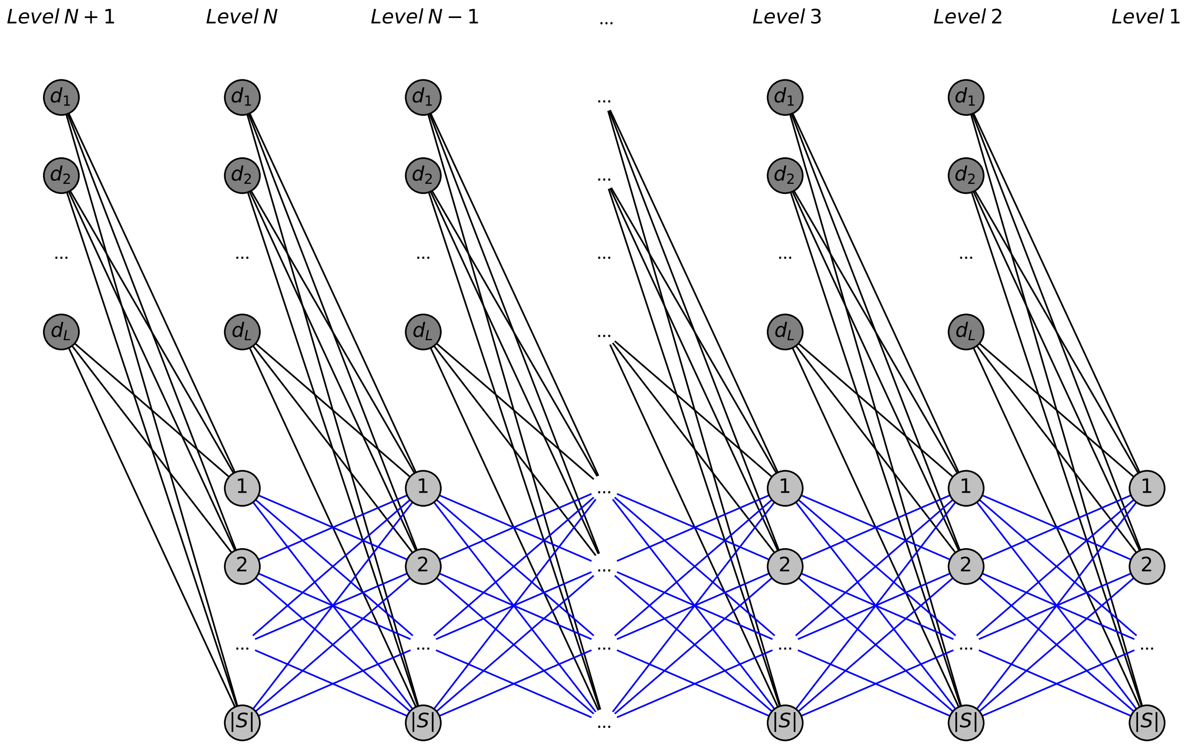

Figure 1: Multi-level network for offline problems

$$\text{Minimize} \quad W \tag{1}$$

S.t.

$$\sum_{s=1}^{|S|} \pi^{l}_{d_l s} y^{rl}_{d_l s} + \sum_{s=1}^{|S|} \sum_{s'=1, s' \neq s}^{|S|} \pi^{l}_{ss'} y^{rl}_{ss'} \leq W, \quad l = \{1, ..., L\} \tag{2}$$

$$\sum_{l=1}^{L} \sum_{r=1}^{N} x^{rl}_{s} = 1, \quad s = \{1, ..., |S|\} \tag{3}$$

$$\sum_{s=1}^{|S|} x^{1l}_{s} = 1, \quad l = \{1, ..., L\} \tag{4}$$

$$\sum_{r=1}^{N} \sum_{s=1}^{S} y^{rl}_{d_l s} = 1, \quad l = \{1, ..., L\} \tag{5}$$

$$\sum_{s'=1, \neq s}^{|S|} y^{rl}_{ss'} = x^{r+1l}_{s}, \quad l = \{1, ..., L\}, s = \{1, ..., |S|\}, r = \{1, ..., N-1\} \tag{6}$$

$$y^{rl}_{d_l s'} + \sum_{s=1, \neq s'}^{|S|} y^{rl}_{ss'} = x^{rl}_{s'} \quad l = \{1, ..., L\}, s' = \{1, ..., |S|\}, r = \{1, ..., N-1\} \tag{7}$$

$$y^{Nl}_{d_l s} = x^{Nl}_{s} \quad s = \{1, ..., |S|\}, l = \{1, ..., L\} \tag{8}$$

$$x_s^{rl} \in \{0,1\} \quad s \in \{1,...,|S|\}, l \in \{1,2,...,L\}, r \in \{1,...,N\} \tag{9}$$

$$y_{ss'}^{rl} \in \{0,1\} \quad s \in \{1,...,|S|\}, s' \in \{1,...,|S|\} \setminus s, l \in \{1,2,...,L\}, r \in \{1,...,N\} \tag{10}$$

$$W \in \mathbb{R} \tag{11}$$

In the given model for the makespan minimization problem, there are two main set of variables: x_s^{rl} shows if node s is served by SAR team l in level r or not and $y_{ss'}^{rl}$ shows if SAR team l goes from node s to in level $r+1$ to node s' in level r or not. Since the objective is to minimize the maximum latency (the makespan), the total time spent by SAR team l for both routing and servicing the critical nodes can be calculated by $\sum_{s=1}^{|S|} \pi_{d_l s}^l y_{d_l s}^{rl} + \sum_{s=1}^{|S|} \sum_{s'=1, s' \neq s}^{|S|} \pi_{ss'}^l y_{ss'}^{rl}$. Hence, the objective function and Constraints (2) set this criterion.

According to Constraints (3), each of the critical nodes should be visited with exactly one SAR team and only in one of the levels. Constraints (4) provide that since each team should service at least one critical node, at level 1, each of the teams should visit one of the critical nodes. Constraints (5) ensure all the teams leave their depot to service a critical node. They can leave the depot in any of the levels from 2 to $N+1$. Constraints (6) link the y and x variables and the levels for outgoing edges from a critical node to other critical nodes. Constraints (7) link the y and x variables and the levels for the incoming edges to a critical node either from the depot or other critical nodes. When there is an active node in level N of team l, Constraints (8) ensure that the corresponding SAR team visits it directly from the depot.

4.1.2. MIP model for the offline weighted latency minimization problem

In the offline weighted latency minimization problem, the weighted latency of a critical node is calculated as the time taken from time zero until the end of the SAR operation at that node multiplied by the ratio of the number of victims at the node to the total number of victims at all the critical nodes. For P_{WL}, we only change the objective function from makespan to weighted latency and use the similar constraints given from (3) to (10). Using the weighted latency objective, we ensure that nodes with more victims have a higher priority.

5. Lower bounds on the competitive ratio of deterministic strategies for P_M and P_{WL}

In this section, we propose two lower bounds on the competitive ratio of deterministic online strategies for both P_M and P_{WL}.

5.1. An immediate lower bound

A closely related problem to P_M and P_{WL} is the *online multi-agent k-CTP with complete communication* (Zhang et al., 2013). In this problem, an undirected connected graph is given with a source node O and a destination node D, together with non-negative edge costs. The traveling agents who can communicate are initially positioned at O. There are k blocked edges which are not known to the agents a priori. A blocked edge cannot be traversed and is learned when an agent arrives at one of its end-nodes. This information is immediately communicated to the other agents. The objective is to find an online strategy such that at least one of the agents finds a route from O to D with minimum travel cost.

Zhang et al. (2013) proved a lower bound of $2\lfloor \frac{k}{L} \rfloor + 1$ on the competitive ratio of deterministic strategies for the online multi-agent k-CTP with complete communication, where k denotes the number of blocked edges and L represents the number of traveling agents. One can verify that the online multi-agent k-CTP with complete communication is a special case of P_M and P_{WL} with only one critical node (D), where the required amount of SAR effort on D is negligible in comparison to edge traveling costs.

Lemma 5.1. *No deterministic strategy achieves a competitive ratio less than* $2\lfloor \frac{k}{L} \rfloor + 1$ *for* P_M *and* P_{WL}.

Proof. Since the multi-agent k-CTP with complete communication is a special case of P_M and P_{WL}, the lemma follows. □

In the next section, we derive a slightly tighter lower bound on the competitive ratio of deterministic online strategies for P_M and P_{WL} by analyzing instances of the problems with more than one critical node.

5.2. An improved lower bound

Lemma 5.2. *No deterministic strategy achieves a competitive ratio less than* $2\lceil \frac{k}{L} \rceil + 1$ *for* P_M *and* P_{WL}, *where* k *denotes the number of blocked edges and* L *represents the number of SAR teams.*

Proof. We show that for an arbitrary deterministic online strategy ALG applied to P_M and P_{WL}, there exists at least one instance of inputs, Γ, such that ALG cannot achieve a competitive ratio better than $2\lceil \frac{k}{L} \rceil + 1$ on Γ.

- **Description of Γ.** We consider the graph in Figure 2, where

$$V = \{d_1\} \cup \{\lambda_i^l | l \in \{1, 2, ..., L\}, i \in \{1, 2, ..., k+1\}\} \cup \{v_1, v_2, ..., v_L\}$$

and

$$E = \{(d_1, \lambda_i^l), (\lambda_i^l, v_l) | l \in \{1, 2, ..., L\}, i \in \{1, 2, ..., k+1\}\}.$$

Let the traveling time of the edges (d_1, λ_i^l) be one for $l = 1, 2, ..., L$ and $i = 1, 2, ..., k+1$. Let the traveling time of the remaining edges be ϵ. We assume that the number of critical nodes equals the number of SAR teams. Also, we assume that $S = \{v_1, v_2, ..., v_L\}$ is the set of critical nodes and $\Lambda = \{\lambda_i^l | l \in \{1, 2, ..., L\}, i \in \{1, 2, ..., k+1\}\}$ is the set of intermediate nodes. We set the required amount of rescue effort at critical node $v_s \in S$ equal to ϵ, i.e. $h_s = \epsilon$. We assume that the SAR teams $T_1, T_2, ..., T_L$ are initially positioned at the depot node d_1, i.e. $D = \{d_1\}$. We let the SAR operation rate of SAR team T_l be one for $l = 1, 2, ..., L$, i.e. $r_l = 1$. Let P_i^l be the path from d_1 to v_l via the intermediate node λ_i^l for $l = 1, 2, ..., L$ and $i = 1, 2, ..., k+1$. Also let $A = \{P_i^l | l \in \{1, 2, ..., L\}, i \in \{1, 2, ..., k+1\}\}$, i.e. A is the set of $L(k+1)$ paths from d_1 to v_l $(l = 1, 2, ..., L)$. For an arbitrary deterministic online strategy ALG, we let $A^{'} \subset A$ be the set of $L\lfloor \frac{k}{L} \rfloor$ paths (from d_1 to v_l) in A which are taken earlier than the other $L(K+1) - L\lfloor \frac{k}{L} \rfloor$ paths in A by the SAR teams. We assume that the k blocked edges belong to $E^{'} = \{(\lambda_i^l, v_l) | l \in \{1, 2, ..., L\}, i \in \{1, 2, ..., k+1\}\}$, i.e. $B \subset E^{'}$. For ALG, we consider the instance in which all of the paths in $A^{'}$ are blocked.

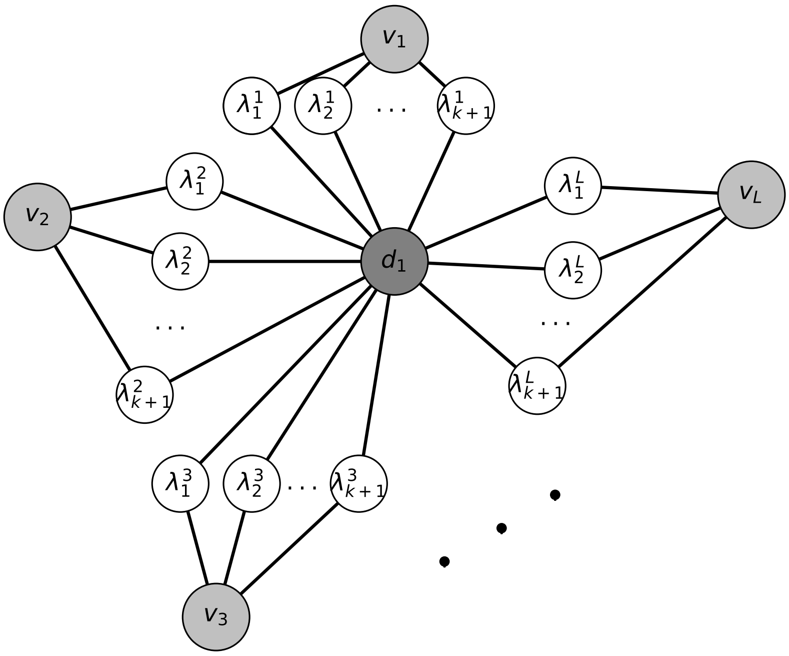

Figure 2: Topology of the instances of P_M and P_{WL} used for the proof of Lemma 5.2

- **Computation of the makespan and the weighted total latency found by ALG on instance Γ.** Note that we consider the instance in which all of the paths in $A^{'}$ are blocked. Hence, in ALG time of at least $2\lfloor\frac{k}{L}\rfloor$ will be taken from time zero until the L number of SAR teams who are initially at d_1 traverse the $L\lfloor\frac{k}{L}\rfloor$ paths in $A^{'}$, encounter $L\lfloor\frac{k}{L}\rfloor$ blocked edges, and arrive back at d_1. To draw a smallest competitive ratio, we assume that ALG is such that the SAR teams have found $L\lfloor\frac{k}{L}\rfloor$ blocked edges and are positioned at d_1 at time $2\lfloor\frac{k}{L}\rfloor$. We consider two cases.

  - **Case 1.** $\lfloor\frac{k}{L}\rfloor < \frac{k}{L} < \lceil\frac{k}{L}\rceil$. Considering the instance described above, for ALG, none of the critical nodes have been visited by time $2\lfloor\frac{k}{L}\rfloor$. Since $\lfloor\frac{k}{L}\rfloor < \frac{k}{L}$, there exists at least one blocked edge in the graph at time $2\lfloor\frac{k}{L}\rfloor$. For ALG, we consider the instance in which one of the SAR teams encounters a blocked edge before visiting a critical node after time $2\lfloor\frac{k}{L}\rfloor$, i.e. the SAR team who encounters the $(L\lfloor\frac{k}{L}\rfloor + 1)^{th}$ blocked edge has to backtrack to d_1 before visiting a critical node. Thus, at least one of the critical nodes cannot be serviced earlier than time $2\lfloor\frac{k}{L}\rfloor + 3 + 2\epsilon$. We let $v_{s^*} \in S$ be the node which cannot be serviced earlier than time $2\lfloor\frac{k}{L}\rfloor + 3 + 2\epsilon$. We set $N_s = 1$ for $v_s \in S - \{v_{s^*}\}$, and $N_{s^*} = M$ for v_{s^*}. Therefore, a time of at least $2\lfloor\frac{k}{L}\rfloor + 3 + 2\epsilon = 2\lceil\frac{k}{L}\rceil + 1 + 2\epsilon$ will be taken for ALG to visit and satisfy all of the critical nodes. Hence the makespan of ALG is at least $2\lceil\frac{k}{L}\rceil + 1 + 2\epsilon$. Also, the total weighted latency of ALG is at least $\frac{M}{M+L-1}(2\lfloor\frac{k}{L}\rfloor+3+2\epsilon)+\frac{(L-1)}{M+L-1}(2\lfloor\frac{k}{L}\rfloor+1+2\epsilon) = \frac{M}{M+L-1}(2\lceil\frac{k}{L}\rceil+1+2\epsilon)+\frac{(L-1)}{M+L-1}(2\lfloor\frac{k}{L}\rfloor+1+2\epsilon)$.
  - **Case 2.** $\lfloor\frac{k}{L}\rfloor = \frac{k}{L} = \lceil\frac{k}{L}\rceil$. Note that for ALG, time of at least $2\lfloor\frac{k}{L}\rfloor = 2\lceil\frac{k}{L}\rceil$ will be taken from time zero until the L number of SAR teams who are initially at d_1 explore the k blocked edges in the graph and arrive back at d_1. Next, it takes a time of at least $1 + 2\epsilon$ for the SAR teams to arrive at the critical nodes and complete the required amount of SAR effort on them. Thus, the makespan of ALG would be at least $2\lceil\frac{k}{L}\rceil + 1 + 2\epsilon$. Note that, the latency of all of the critical

nodes equals $2\lfloor \frac{k}{L} \rfloor + 1 + 2\epsilon = 2\lceil \frac{k}{L} \rceil + 1 + 2\epsilon$. Thus, the total weighted latency of ALG is at least $2\lceil \frac{k}{L} \rceil + 1 + 2\epsilon$.

- **Computation of the makespan and the weighted total latency of the offline optimum.** Since the number of SAR teams and the number of critical nodes are equal, in the offline optimum it takes a time of $1 + 2\epsilon$ for the L SAR teams to arrive at the L critical nodes and complete the required amount of SAR effort on them. Therefore, the makespan and the total weighted latency of the offline optimum is $1 + 2\epsilon$.

The lemma follows for P_M when ϵ approaches zero. The lemma follows for P_{WL} when ϵ approaches zero and M approaches $+\infty$. □

6. Description of the strategies

In this section, we present two deterministic online strategies for both P_M and P_{WL}.

6.1. MIP-based strategy

We propose a deterministic online strategy for both P_M and P_{WL} which utilizes the solution of an offline MIP model (the model in Section 4.1.1 for P_M and the model in Section 4.1.2 for P_{WL}) to partition the critical nodes into clusters such that each cluster is assigned to exactly one SAR team. We call this strategy the *MIP-based strategy*. Before we describe our strategy, we need to present the following definitions.

- **An unassigned critical node.** We say a critical node $v_s \in S$ is *unassigned* if the required amount of SAR effort at v_s is not observed and it does not belong to the cluster of any of the SAR teams. We denote the set of unassigned critical nodes by S^0.
- **An observed critical node.** We say a critical node $v_s \in S$ is *observed*, if the required amount of SAR effort at v_s is observed but it is not determined which SAR team should perform the required SAR effort at v_s. We show the set of observed critical nodes by S^1, i.e. $S^0 \cap S^1 = \emptyset$.
- **A planned critical node.** We say a critical node $v_s \in S$ is *planned*, if the required amount of SAR effort at v_s is observed and it is determined which SAR team should perform the required SAR effort at v_s. We show the set of observed critical nodes by S^2, i.e. $S^0 \cap S^2 = \emptyset$, and $S^1 \cap S^2 = \emptyset$.
- **An idle SAR team.** We say a team $T_l \in T$ is *idle* if it is static at its current location and does not perform any SAR operation. We show the set of idle teams by $T^1 \subset T$, i.e. $T^1 = T$ at the beginning.
- **A directed SAR team.** We say that a team $T_l \in T$ is *directed* to a node $v_s \in S$ (i.e., $v_s \notin S^1 \cup S^2$), if T_l is responsible from taking the shortest path between their current location and v_s to observe the required amount of rescue effort at v_s. We show the set of directed teams by $T^2 \subset T$, i.e. $T^1 \cap T^2 = \emptyset$. When a team in T^2 observes the required amount of SAR effort at its directed node, it is removed from T^2 and added to T^1.

- **An appointed SAR team.** We say that a team $T_l \in T$ is *appointed* to a planned node $v_s \in S^2$, if T_l is given the responsibility to take the shortest path between its current location and v_s. We show the set of appointed teams by $T^3 \subset T$, i.e. $T^1 \cap T^3 = \emptyset$ and $T^2 \cap T^3 = \emptyset$.
- **A busy SAR team.** We say that a team $T_l \in T$ is *busy* at a planned node $v_s \in S^2$, if T_l is performing SAR operation at v_s. We show the set of busy teams by $T^4 \subset T$, i.e. $T^1 \cap T^4 = \emptyset$, $T^2 \cap T^4 = \emptyset$, $T^3 \cap T^4 = \emptyset$, and $T^1 \cup T^2 \cup T^3 \cup T^4 = T$. When a team in T^3 arrives at its appointed node, it is removed from T^3 and added to T^4.
- **The blockage factor.** When a team encounters a blocked edge, the strategy for all the teams is updated. In such cases, a directed or appointed team may change its current route and incur a higher traveling cost. To take this into account, we consider the *blockage factor* (β). Note that β is defined as a pre-determined constant which is greater than one.

Our MIP-based strategy ALG_{MIP} utilizes the given incomplete information on the inputs and disregards the unknown information as follows. At the beginning, ALG_{MIP} assumes that there is no blocked edge in the graph and the required amount of rescue effort at all of the critical nodes is zero. Then, it solves a MIP model (the model in Section 4.1.1 for P_M and the model in Section 4.1.2 for P_{WL}) in order to partition the critical nodes into clusters such that each cluster is assigned to exactly one SAR team. Note that at this point, all the SAR teams are idle. Next, each idle team is directed to the closest critical node in its cluster. If a blocked edge is found, ALG_{MIP} removes the blocked edge from the graph and updates the strategy for all the SAR teams. When a team arrives at its directed node, the team becomes idle and ALG_{MIP} appoints a SAR team to the node. When a team arrives at its appointed node, the team becomes busy and performs the required amount of rescue effort at the node. When a team completes the required amount of rescue effort at a critical node, it becomes idle and may be directed or appointed to another critical node if possible. Below, we formally present the MIP-based strategy.

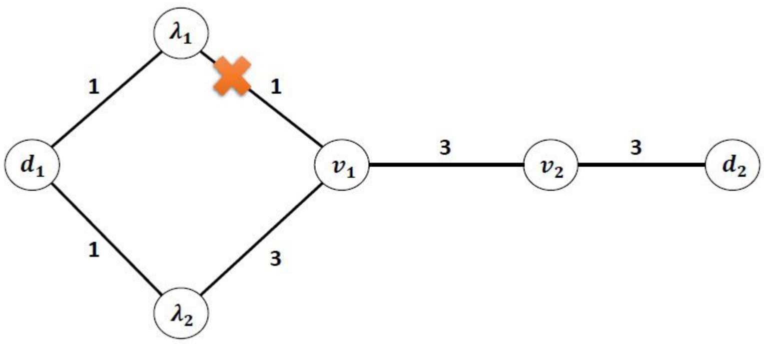

Figure 3: An illustrative example for the MIP-based strategy

The MIP-based strategy:

- **Input:** Undirected graph $G = (V, E)$ together with the edge traveling costs, set of critical nodes $S \subset V$, set of SAR teams $T = \{T_l | l = 1, 2, 3, ..., L\}$, r_l SAR operation rate of T_l, set of depot nodes D.

- **Initialization:** Define $S^0 \subset S$ as the set of *unassigned* critical nodes and set $S^0 = S$. Define $S^1 \subset S$ as the set of *observed* critical nodes and set $S^1 = \emptyset$. Define $S^2 \subset S$ as the set of *planned* critical nodes and set $S^2 = \emptyset$, i.e. $S^0 \cap S^1 = \emptyset$, $S^0 \cap S^2 = \emptyset$, and $S^1 \cap S^2 = \emptyset$. Define $T^1 \subset T$ as the set of *idle* SAR teams and set $T^1 = T$ at the beginning. Define $T^2 \subset T$ as the set of *directed* SAR teams and set $T^2 = \emptyset$, i.e. $T^1 \cap T^2 = \emptyset$. Define $T^3 \subset T$ as the set of *appointed* SAR teams and set $T^3 = \emptyset$, i.e. $T^1 \cap T^3 = \emptyset$ and $T^2 \cap T^3$. Define $T^4 \subset T$ as the set of *busy* SAR teams and set $T^4 = \emptyset$, i.e. $T^1 \cap T^4 = \emptyset$, $T^2 \cap T^4 = \emptyset$, $T^3 \cap T^4 = \emptyset$, and $T = T^1 \cup T^2 \cup T^3 \cup T^4$. Let $\beta > 1$ be the blockage factor (we set $\beta = 1.5$ in our implementation).

- **Step 1.** Set $h_s = 0$ for $v_s \in S$, i.e. set the required amount of rescue effort at critical nodes equal to zero. If the problem is P_M, solve an exact MIP model for makespan minimization which is presented in Section 4.1.1. If the problem is P_{WL}, solve an exact MIP model for total weighted latency minimization which is presented in Section 4.1.2. Set $S^0 = \emptyset$ and let U_l be the set of critical nodes which are assigned to the SAR team $T_l \in T$ according to the MIP model. Go to Step 2.

- **Step 2.** If $T^1 = \emptyset$ or $U_l \cup S^0 = \emptyset$ for all $T_l \in T^1$, to update the strategy for all the teams go to Step 3. Otherwise, arbitrarily choose T_{l^*} among the SAR teams in T^1 whose $U_l \cup S^0 \neq \emptyset$. Find the nearest critical node $v_{i^*} \in U_{l^*} \cup S^0$ to the current location of T_{l^*} and *direct* T_{l^*} to v_{i^*}. Set $U_{l^*} = U_{l^*} - \{v_{i^*}\}$, $S^0 = S^0 - \{v_{i^*}\}$, $T^2 = T^2 \cup \{T_{l^*}\}$, and $T^1 = T^1 - \{T_{l^*}\}$. Then, go to the beginning of Step 2.

- **Step 3.** For $T_l \in T^2 \cup T^3 \cup T^4$:

  - If $T_l \in T^2$, make T_l responsible for traversing the shortest path from its current location to its directed critical node.
  - If $T_l \in T^3$, make T_l responsible for traversing the shortest path from its current location to its appointed critical node.
  - If $T_l \in T^4$, make T_l responsible for performing SAR operation on its appointed critical node.

  Let the teams in $T_l \in T^2 \cup T^3 \cup T^4$ begin their assigned responsibilities simultaneously;

  - If one of the teams encountered a blocked edge e, set $E = E - \{e\}$, and to update the strategy for all of the teams go to the beginning of Step 3.
  - If a team $T_l \in T^2$ arrived at its directed node v_s, observe h_s (the required amount of rescue effort at v_s), set $S^1 = S^1 \cup \{v_s\}$, $T^2 = T^2 - \{T_l\}$, $T^1 = T^1 \cup \{T_l\}$, and to update the strategy for all the teams go to Step 4.
  - If a team $T_l \in T^3$ arrived at its appointed node $v_s \in S_2$, $T^3 = T^3 - \{T_l\}$, $T^4 = T^4 \cup \{T_l\}$, and to update the strategy for all the teams go to the beginning of Step 3.
  - If a team $T_l \in T^4$ completed the required amount of rescue effort at its planned node $v_s \in S_2$, set $T^4 = T^4 - \{T_l\}$, and $T^1 = T^1 \cup \{T_l\}$. If the required amount of rescue effort at all the critical nodes is fulfilled, stop. Otherwise, to update the strategy for all of the teams go to the beginning of Step 2.

- **Step 4.** Choose $v_{i^*} \in S^1$ arbitrarily. Let $c_l^{i^*}$ be the travel time of the shortest path between the current location of $T_l \in T$ and v_{i^*}. Compute $\pi_l = \frac{h_{i^*}}{r_l} + \beta(c_l^{i^*})$ for $T_l \in T^1 \cup T^2$. Let $T_{l^*} \in T^1 \cup T^2$ be the SAR team whose π_l is minimum (choose the team which is closer to v_{i^*} in case of tie). If T_{l^*} was previously directed to a critical node $v_s \neq v_{i^*}$ (if $T_{l^*} \in T^2$), set $S^0 = S^0 \cup \{v_s\}$. *Appoint* T_{l^*} to v_{i^*}, set $S^1 = S^1 - \{v_{i^*}\}$, $S^2 = S^2 \cup \{v_{i^*}\}$, $T^1 = T^1 - \{T_{l^*}\}$, $T^2 = T^2 - \{T_{l^*}\}$, $T^3 = T^3 \cup \{T_{l^*}\}$. If $S^1 = \emptyset$, to update the strategy for all the teams go to Step 2. Otherwise, go to the beginning of Step 4.

Remark 6.1. *Consider a constructive version of the MIP-based strategy in which whenever a team $T_l \in T^2$ arrives at its directed critical node $v_s \in S$ (Step 3) it is immediately appointed to v_s, i.e. in the constructive version of the MIP-based strategy Step 4 is eliminated. We added Step 4 to the constructive version of the MIP-based strategy to improve the quality of the solution. Hereafter, we call Step 4, the improvement step of the MIP-based strategy.*

Remark 6.2. *We tested our online strategies on various instances with up to 30 critical nodes. The MIP-based strategy could not provide the optimal solutions of the larger instances. This is because the MIP-based strategy requires solving the offline MIP model within the algorithm. However, this limitation is remedied in the greedy strategy which does not require solving a MIP model at all. Rather, the main purpose of having the MIP-based strategy is to benchmark it with the fast greedy algorithm.*

Table 2: Summarizing the illustrative example for the MIP-based strategy

Time	**Event**	T^1	T^2	T^3	T^4
0	T_1 and T_2 are idle	$\{T_1, T_2\}$	$\emptyset$	$\emptyset$	$\emptyset$
0	T_1 is directed to v_1 and T_2 is directed to v_2	$\emptyset$	$\{T_1, T_2\}$	$\emptyset$	$\emptyset$
1	blocked edge (λ_1, v_1) is found by T_1	$\emptyset$	$\{T_1, T_2\}$	$\emptyset$	$\emptyset$
3	h_2 is observed at v_2 by T_2	$\{T_2\}$	$\{T_1\}$	$\emptyset$	$\emptyset$
3	T_1 is appointed to v_2	$\{T_2\}$	$\emptyset$	$\{T_1\}$	$\emptyset$
3	T_2 is directed to v_1	$\emptyset$	$\{T_2\}$	$\{T_1\}$	$\emptyset$
6	h_1 is observed at v_1 by T_2	$\{T_2\}$	$\emptyset$	$\{T_1\}$	$\emptyset$
6	T_2 is appointed to v_1	$\emptyset$	$\emptyset$	$\{T_1, T_2\}$	$\emptyset$
6	T_2 is busy at v_1	$\emptyset$	$\emptyset$	$\{T_1\}$	$\{T_2\}$
9	T_1 is busy at v_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{T_1, T_2\}$
18	T_2 finishs	$\{T_2\}$	$\emptyset$	$\emptyset$	$\{T_1\}$
19	T_1 finishes	$\{T_1, T_2\}$	$\emptyset$	$\emptyset$	$\emptyset$

As an illustrative example, consider the instance given in Figure 3, where $S = \{v_1, v_2\}$ is the set of critical nodes, $D = \{d_1, d_2\}$ is the set of depot nodes, $\Lambda = \{\lambda_1, \lambda_2\}$ is the set of intermediate nodes, and the numbers on the edges represent their travel times. Let $T = \{T_1, T_2\}$ be the set of SAR teams. We assume that T_1 and T_2 are initially positioned at d_1 and d_2, respectively. Also we assume that $r_1 = 3$ and $r_2 = 1$, where r_j $(j \in \{1, 2\})$ is the SAR operational rate of T_j. We assume that the edge (λ_1, v_1) is blocked and this is not known to T_1 and T_2 at the beginning. When Step 2 of the MIP-based strategy is implemented, T_1 and T_2

are directed to v_1 and v_2, respectively. At time 1, T_1 arrives at λ_1 and finds the edge (λ_1, v_1) blocked. Next, T_1 and T_2 are directed to v_1 and v_2, respectively. At time 3, T_2 arrives at v_2 and h_2 (the required amount of rescue effort at v_2) is revealed. Let $h_2 = 30$. Note that at time 3, T_1 is at λ_2. Here, the MIP-based strategy enters Step 4 and computes $\pi_1 = \frac{30}{3} + 1.5(6) = 19$ and $\pi_2 = \frac{30}{1} + 1.5(0) = 30$. Since $\pi_1 < \pi_2$, T_1 is appointed to v_2 and T_2 is directed to v_1. T_2 arrives at v_1 at time 6 and $h_1 = 12$ is revealed. T_2 completes the required amount of rescue effort at v_1 at time $6 + 12 = 18$. Also, T_1 completes the required amount of rescue effort at v_2 at time $3 + 6 + 10 = 19$. Table 2 summarizes the illustrative example described above.

In the next section, we propose another strategy for P_M and P_{WL} which does not depend on solving a mathematical model.

6.2. Greedy strategy

In this section, we introduce another deterministic online strategy for both P_M and P_{WL}. Since the routing choices are greedy in our strategy, we call it the *Greedy strategy*. Our Greedy strategy, ALG_{greedy}, is similar to our MIP-based strategy with one main difference. Instead of solving a MIP model to cluster the critical nodes, ALG_{greedy} directs each SAR team to an unassigned critical node closest to its current location. Hence, Step 1 of the MIP-based strategy is eliminated in the Greedy strategy. The further steps of ALG_{greedy} are very similar to the corresponding steps in ALG_{MIP}. Below, we formally describe our deterministic online greedy strategy.

The greedy strategy:

- **Input:** The input of the MIP-based strategy hold.
- **Initialization.** The initialization of the MIP-based strategy hold.
- **Step 1.** If $T^1 = \emptyset$ or $S^0 = \emptyset$, to update the strategy for all the teams go to Step 2. Otherwise, choose $T_{l^*} \in T^1$ arbitrarily. Find the nearest critical node $v_{i^*} \in S^0$ to the current location of T_{l^*} and *direct* T_{l^*} to v_{i^*}. Set $S^0 = S^0 - \{v_{i^*}\}$, $T^2 = T^2 \cup \{T_{l^*}\}$, and $T^1 = T^1 - \{T_{l^*}\}$. Then, go to the beginning of Step 1.
- **Step 2.** Similar to Step 3 of the MIP-based strategy with the following difference. If a team $T_l \in T^2$ arrived at its directed node $v_s \in S^0$, observe h_s (the required amount of rescue effort at v_s), set $S^1 = S^1 \cup \{v_s\}$, $T^2 = T^2 - \{T_l\}$, $T^1 = T^1 \cup \{T_l\}$, and to update the strategy for all the teams go to Step 3.
- **Step 3.** Similar to Step 4 of the MIP-based strategy.

Remark 6.3. *Similar to Step 4 of the MIP-based strategy which is called the improvement step of the MIP-based strategy, we call Step 3 of the Greedy strategy the improvement step of the Greedy strategy.*

Remark 6.4. *The required amount of SAR effort at the nodes are not known a priori in online problems. As a result, the optimal solutions of the online problems cannot be obtained using an enumeration procedure, even if the number of critical nodes that can be serviced within 48 hours is reasonably small. Although* *offline problems can be solved by an enumeration procedure for small instances, we chose not to implement an enumeration procedure since the main concentration of this study is to propose efficient online solutions for the online problems and to show their average computational performance in addition to the theoretical worst-case bounds. Therefore, we solved the offline problem using a MIP model instead of an enumeration procedure.*

7. Computational Results

In this section, we present the results of our experiments to investigate the performance of the proposed online strategies for P_M and P_{WL} in comparison to the MIP models suggested for the offline problems. We mainly focus on run times of the strategies and how far solutions they yield compared to the offline optimal solutions. We compare the performance of two alternative strategies.

7.1. Data Generation

Our experiments are conducted on 10 randomly generated networks and different sets of parameters. In the instances, each generated graph is a random network in a 300×300 square with 500 nodes. There is an edge between a pair of nodes if their Euclidean distance is less than 30. This is to represent that the maximum travel time between a pair of nodes is assumed to be 30 minutes. In these instances, the average number of edges is slightly more than 3500. This corresponds to an average degree of 14 over the nodes. In comparison with various road networks that are generated from real data in the literature (see (Rawls & Turnquist, 2010), (Ajam et al., 2019), (Akbari & Salman, 2017a), and (Akbari & Salman, 2017b)) where the average degree of the nodes in all of the cases is less than five, these networks are relatively dense and make the instances more challenging.

We tested each of the networks under various settings. We randomly selected blocked edges but made sure to keep the network connected. We generated instances where 10, 25 and 40 percent of the edges are blocked, to represent instances under different disaster impact, corresponding to minor, moderate and severe conditions, respectively. We refer to the percentage of edge blockage by PEB in the tables given in the following sections.

For each of the networks, under different percentage of blocked edges mentioned above, we also tested all of the instances with the number of critical nodes being 20, 25 and 30. Here the goal is to see how the models and the strategies scale to the number of critical nodes, which is shown by $|S|$ in the following tables. The number of victims in each critical node is a randomly generated number between 1 and 30. Moreover, the average SAR operation time for each of these victims is assumed to be a random value between 30 and 180 minutes.

For any of the instances under different disaster severity cases and with different number of critical nodes, we tested them with 3, 6 and 9 SAR teams. The number of SAR teams is shown by L in the following tables. The SAR teams are categorized into 3 groups; the more equipped group with a capacity of 30 SAR operation units in an hour, the moderately equipped SAR team with a capacity of 20 SAR operation units in an hour and finally the least equipped SAR team with a capacity of 10 SAR operation units in an hour. In each of the instances, the number of SAR teams from these groups are equal to each other. That is why the number of teams is decided to be a multiplier of 3. Note that each of the 10 randomly generated networks is tested $3 \times 3 \times 3 = 27$ times, resulting in 270 instances in total.

Remark 7.1. *The optimal offline solutions could not be obtained for the instances with more than 30 critical nodes. Hence, the competitive ratio of the greedy strategy could not be computed on these instances, even though the greedy strategy which does not require solving a MIP model is very fast and thus handles very large instances.*

7.2. Computational Results

The computational experiments were conducted with Python 2.7.16 using Gurobi 8.0 on Intel Core(TM) i5-7200U CPU @ 2.50GHz 2.71GHz (two processors) computer with 8 GB RAM, running under the Windows 10 operating system. In the following, we first give the results of the makespan minimization problem, P_M and then give the results of the weighted latency minimization problem, P_{WL}.

7.2.1. Results for the makespan minimization problem

At first, we tested the offline model presented in Section 4.1.1 for P_M in a 30 minute time limit. Then we tested both MIP-based and Greedy strategies on the same instances. These results are given in Table 3.

In Table 3, the column denoted by $|S|$ shows the number of critical nodes in that instance. The PEB column shows the impact of the disaster, which is categorized in three groups as minor, moderate and severe, corresponding to 10, 25 and 40 percent edge blockage in the network, respectively. The column denoted by L gives the number of SAR teams in that instance. As it is stated above, first the offline problem was solved. A run time limit of 1800 seconds was given to the solver and the run time reported in the tables are given in seconds as well. As it can be seen in Table 3, run time increases as the number of critical nodes increases. For instance, while the optimal solutions to all of the tested instances with 20 critical nodes were found within the time limit, none of the instances were solved optimally in the given time limit with 30 critical nodes and 6 or 9 teams. However, the average optimality gap for the unsolved instances remained under 10%. For the instances whose offline models were not solved optimally within the time limit, we used the obtained lower bounds (within the time limit) for the objective function values of the offline problems while calculating the competitive ratios.

After solving the offline problem, we tested both the MIP-based and the Greedy strategies. For each of these strategies, three columns are reported in Table 3. The Avg runtime column shows the average time in seconds that the solver spent on finding the solution. The Improvement % column shows the improvement percentage resulting from adding the improvement step to each of the strategies. Since the nature of the problem is online, there is no guarantee that either of the strategies with the improvement step obtains a better solution. In these experiments, we set the blockage factor as $\beta = 1.5$ based on our preliminary tests. We note that if the value of β decreases, the possibility of having negative improvement increases and if the value of β increases, the possibility of having positive improvement decreases and as a result, the average improvement decreases. The CR column shows the average competitive ratio of the online strategies on the tested instances. The underlined entries in the CR column are the better ones between the tested strategies. Note that the competitive ratio is different from optimality gap which is used to assess the performance

			Offline Makespan	MIP Based Strategy			Greedy Strategy		
|S|	PEB	L	Avg runtime	Avg runtime	Improvement? %	CR	Avg runtime	Improvement? %	CR
20	Minor	3	25.42	19.89	21.81	1.23	0.72	1.63	1.22
		6	250.14	64.28	17.53	1.28	0.87	8.71	1.38
		9	464.86	12.95	26.01	1.17	1.05	12.40	1.47
	Moderate	3	34.74	25.02	11.38	1.28	0.71	6.00	1.20
		6	814.60	37.18	17.29	1.33	0.89	7.96	1.37
		9	129.80	7.76	34.48	1.38	1.03	14.14	1.28
	Severe	3	37.60	24.60	5.47	1.14	0.78	1.17	1.35
		6	378.00	122.60	6.25	1.45	0.92	5.05	1.38
		9	257.07	14.00	36.14	1.28	1.06	14.95	1.56
25	Minor	3	148.49	90.52	24.18	1.23	0.97	2.42	1.11
		6	1521.20	767.80	4.75	1.28	1.19	13.24	1.34
		9	1795.20	50.72	34.69	1.36	1.50	6.44	1.48
	Moderate	3	125.82	362.40	5.60	1.24	0.86	7.38	1.23
		6	1800	1303.40	24.23	1.40	1.19	13.83	1.35
		9	1651.00	270.00	23.18	1.22	1.39	14.83	1.36
	Severe	3	107.20	169.20	10.79	1.32	1.04	-0.36	1.30
		6	1800	1143.20	12.20	1.25	1.31	6.59	1.51
		9	1671.40	56.60	26.62	1.35	1.33	-0.64	1.52
30	Minor	3	548.33	808.24	23.39	1.22	1.57	1.95	1.19
		6	1800	1588.20	24.71	1.21	1.45	6.86	1.33
		9	1800	1331.00	28.74	1.28	1.62	7.97	1.40
	Moderate	3	207.20	458.20	17.73	1.41	1.34	3.49	1.18
		6	1800	1687.20	24.31	1.33	1.52	0.23	1.36
		9	1800	1286.00	22.95	1.27	1.68	0.34	1.29
	Severe	3	466.80	413.80	8.05	1.30	1.32	-2.87	1.25
		6	1800	1676.00	6.00	1.26	1.47	6.89	1.36
		9	1800	1110.00	5.44	1.39	1.94	9.17	1.42

Table 3: Results from solving the makespan minimization problem

of heuristics in offline optimization problems. The competitive ratio of a strategy is the result of dividing the obtained objective function value of the online strategy over the optimal objective function value of the offline strategy.

We can see that while for the Greedy strategy the run time remains under 2 seconds on average for all of the tested instances, it increases significantly for the MIP-based strategy as the number of critical nodes increases. This is a direct result of solving the MIP model to initialize the strategy and this step is responsible for approximately 99% of the run time. The improvement step worked better for the MIP-based strategy compared to the Greedy strategy where some of the improvements turned out to be negative. For the MIP-based strategy, the improvement step performed better in the minor severity case with an average of 22.87% compared to the moderate (20.13%) and severe (13.01%) cases. Finally, the comparison between the average competitive ratios shows that on average the MIP-based strategy performed better when the number of SAR teams increases. In most of the cases with 3 SAR teams, the Greedy strategy performed better with an average of 1.22, while the average competitive ratio for the MIP-based strategy is 1.26. However, in the instances with 6 and 9 SAR teams, the MIP-based strategy performed better with an average competitive ratio of 1.31 for 6 and 1.30 for 9 SAR teams. The corresponding competitive ratios for the Greedy strategy are 1.37 and 1.41 for 6 and 9 SAR teams, respectively.

Although the level of difficulty of an online problem defines what competitive ratios are acceptable for a strategy, the obtained values of the competitive ratio for our strategies show that the introduced strategies are performing well based on the online optimization literature. In some computational studies such as Zhang et al. (2019), which addresses the online minimum latency problem with up to k blocked roads that are discovered upon arriving to a node adjacent to that edge, a strategy with a competitive ratio of below 3 is considered a good strategy.

7.2.2. Results for the weighted latency minimization problem

For P_{WL}, similar to the makespan version, we first solved the offline problem and then tested our MIP-based and Greedy strategies on the same sets of instances.

			Offline Latency	MIP Based Strategy			Greedy Strategy		
|S|	PEB	L	Avg runtime	Avg runtime	Improvement? %	CR	Avg runtime	Improvement? %	CR
20	Minor	3	3.17	3.19	7.35	1.27	0.72	0.41	1.28
		6	2.01	2.20	4.68	1.25	0.88	0.97	1.28
		9	1.96	2.32	4.76	1.24	0.96	3.35	1.29
	Moderate	3	3.05	3.52	0.51	1.36	0.73	2.74	1.34
		6	1.76	2.31	8.82	1.37	0.89	-0.44	1.42
		9	1.83	2.11	17.17	1.25	1.03	0.27	1.28
	Severe	3	3.70	4.91	2.72	1.38	0.80	0.73	1.39
		6	1.47	2.19	0.54	1.35	2.36	1.08	1.38
		9	1.66	1.79	6.08	1.27	1.02	3.82	1.39
25	Minor	3	19.13	23.19	8.16	1.32	0.97	0.34	1.27
		6	4.91	5.58	10.45	1.34	1.17	2.10	1.38
		9	6.07	6.43	9.13	1.27	1.51	4.06	1.27
	Moderate	3	13.40	14.45	3.91	1.39	1.00	1.22	1.39
		6	4.71	7.29	11.10	1.29	1.18	2.03	1.33
		9	4.96	8.43	8.21	1.44	1.39	2.67	1.39
	Severe	3	12.87	20.16	3.45	1.41	1.05	-0.05	1.30
		6	5.97	7.37	6.82	1.31	1.29	-0.64	1.43
		9	4.33	5.42	6.82	1.37	1.24	2.11	1.38
30	Minor	3	64.13	446.44	8.71	1.44	1.57	0.04	1.42
		6	16.80	19.13	4.09	1.35	1.53	-0.99	1.40
		9	14.00	16.80	14.99	1.35	1.62	2.60	1.31
	Moderate	3	40.00	91.04	7.44	1.34	1.32	0.79	1.32
		6	16.96	27.20	12.58	1.34	1.52	-0.59	1.43
		9	15.00	21.50	6.38	1.33	1.68	-0.11	1.35
	Severe	3	103.20	140.80	6.65	1.47	1.41	1.23	1.36
		6	23.37	22.69	7.16	1.32	1.47	2.56	1.34
		9	12.70	20.72	10.30	1.42	1.93	2.79	1.47

Table 4: Results from solving the Weighted Latency Problem

Table 4 contains the same column headings as in Table 3. From Table 4 we can see that unlike P_M, the offline version of the P_{WL} was solved for all the instances within the given time limit. This is mainly because the formulation proposed and tested in Angel-Bello et al. (2017) was aimed to solve the latency objective and not the makespan version. In P_{WL}, the run time increases when the number of critical nodes increases. However, the number of SAR teams has a reverse impact on the run time and as the number of SAR teams increases, the solver seems to be able to find the solutions faster.

Different from P_M, for the weighted latency objective function, the average run time of the MIP-based strategy is not very large as the initialization step which includes solving the MIP model is solved considerably faster. Similar to the makespan results, the average run time for the Greedy strategy is low, i.e. it is under 3 seconds in all of the cases. Looking at the performance of the improvement step, we can see that similar to the results for the makespan minimization problem, the improvement step showed better performance on the initial solutions obtained from the MIP-based strategy compared to those from the Greedy strategy. While the improvement on average is 7.3% for the MIP-based strategy, it is only 1.3% for the Greedy strategy. It is also interesting to see that the average improvement percentages for P_M was considerably higher than the average improvement percentages for P_{WL}. The improvment percentages are 18.6% and 6.3% for the MIP-based strategy and the Greedy strategy, respectively. From Table 4 we also observe that there is no significant difference on the average competitive ratio obtained from the two strategies with 1.34 for the MIP-based and 1.35 for the Greedy strategy.

A comparison between the performance of the MIP-based strategy for the makespan and the weighted latency problems shows that although P_M was noticeably harder to solve with an average run time of 551 seconds over all of the instances, versus the average run time of 33.41 seconds for P_{WL}, the obtained competitive ratio for P_M was 1.29, which is lower than the average competitive ratio of 1.34 for P_{WL}. The same comparison for the Greedy strategy shows that the run time of this strategy is not influenced by the problem and it is under 3 seconds in all of the tested cases, and the average competitive ratio is also very similar for them with being 1.33 for P_M and 1.35 for P_{WL}.

8. Conclusions

We optimized the routing of SAR teams to reach the locations where victims are trapped after a disaster causing a large number of casualties. We also optimized the allocation of heterogeneous teams with different operational capacities to the victim locations that may require different amount of SAR work since different number of victims may exist at the locations and the nature of damage may necessitate different amount of rescue effort. A key characteristic of our work is that online strategies need to be found that will work under uncertainty of rescue times at the locations having trapped victims (nodes of the graph) and the uncertainty of which connections (edges of the graph) may be untraversable. As the strategy runs following a predefined online strategy, uncertainty about these elements are revealed according to the actions of the SAR teams. We believe that this problem characteristic captures the nature of the environment after a disaster. The defined problems are also a new addition to the online optimization literature, since a multi-agent navigation problem on a graph with uncertain edge blockages and having multiple origins and multiple destinations has not been addressed before.

We analyzed the online problem that we proposed by means of two objectives, namely minimization of the time when all rescue operations are completed (makespan) and minimization of the weighted latency of the victims, which is defined to represent the average waiting time of a victim. For these two problems, we provided two lower bounds on the competitive ratio of online deterministic strategies; hence quantifying in terms of problem parameters how much it can be possible in the worst-case to approach the optimal solution when all input data are known at the time of planning.

We devised two online strategies that are valid for both problems; one based on the solution of an offline MIP model, and the other on greedy choices. By extensive computational experiments on randomly generated data, we showed that the Greedy strategy works in a few seconds on large-scale instances with 500 nodes and up to 9 SAR teams but solving the offline makespan model takes more time in some large-sized instances. On the other hand, solving the offline latency problem takes at most a few minutes in large instances.

An open question that remains to be analyzed from a theoretical point of view is deriving the competitive ratio of the proposed strategies as well as the tightness of the proposed lower bounds. From a practical point of view, an analysis of our online strategies on real-life data would be interesting.

References

Ahmadi, M., Seifi, A., & Tootooni, B. (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*, 145 – 163.

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

Akbari, V., & Salman, F. S. (2017a). Multi-vehicle prize collecting arc routing for connectivity problem. *Computers & Operations Research*, *82*, 52 – 68.

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

Alem, D., Clark, A., & Moreno, A. (2016). Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research*, *255*, 187 – 206.

Angel-Bello, F., Cardona-Valdés, Y., & Álvarez, A. (2017). Mixed integer formulations for the multiple minimum latency problem. *Operational Research*, *19*, 369–398.

Anjum, S. S., Noor, R. M., & Anisi, M. H. (2015). Survey on manet based communication scenarios for search and rescue operations. In *2015 5th International Conference on IT Convergence and Security (ICITCS)* (pp. 1–5).

Aslan, E., & Celik, M. (2019). Pre-positioning of relief items under road/facility vulnerability with concurrent restoration and relief transportation. *IISE Transactions*, *51*, 847–868.

Bar-Noy, A., & Schieber, B. (1991). The canadian traveler problem. *SODA '91 Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms*, (pp. 261—270).

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, *34*, 209 – 219.

Bodaghi, B., & Ekambaram, P. (2016). An optimization model for scheduling emergency operations with multiple teams. *International Conference on Industrial Engineering and Operations ManagementAt: Detroit, Michigan, USA*, (pp. 436–442).

Bruni, M., Beraldi, P., & Khodaparasti, S. (2018). A fast heuristic for routing in post-disaster humanitarian relief logistics. *Transportation Research Procedia*, *30*, 304 – 313.

Bulhoes, T., Sadykov, R., & Uchoa, E. (2018). A branch-and-price algorithm for the minimum latency problem. *Computers and Operations Research*, *93*, 66 – 78.

Campbell, A. M., Vandenbussche, D., & Hermann, W. (2008). Routing for relief efforts. *Transportation Science*, *42*, 127—145.

Chen, L., & Miller-Hooks, E. (2012). Optimal team deployment in urban search and rescue. *Transportation Research Part B: Methodological*, *46*, 984 – 999.

Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, *33*, 1 – 27.

Diabat, A., Jabbarzadeh, A., & Khosrojerdi, A. (2019). A perishable product supply chain network design problem with reliability and disruption considerations. *International Journal of Production Economics*, *212*, 125 – 138.

Elci, O., Noyan, N., & Bulbul, K. (2018). Chance-constrained stochastic programming under variable reliability levels with an application to humanitarian relief network design. *Computers & Operations Research*, *96*, 91 – 107.

Ferrer, J. M., Martín-Campo, F. J., Ortuno, M. T., Pedraza-Martinez, A. J., Tirado, G., & Vitoriano, B. (2018). Multi-criteria optimization for last mile distribution of disaster relief aid: Test cases and applications. *European Journal of Operational Research*, *269*, 501 – 515.

Ganz, A., Schafer, J. M., Tang, J., Yang, Z., Yi, J., & Ciottone, G. (2015). Urban search and rescue situational awareness using diorama disaster management system. *Procedia Engineering*, *107*, 349 – 356. Humanitarian Technology: Science, Systems and Global Impact 2015, HumTech2015.

Hoyos, M. C., Morales, R. S., & Akhavan-Tabatabaei, R. (2015). Or models with stochastic components in disaster operations management: A literature survey. *Computers & Industrial Engineering*, *82*, 183 – 197.

Hu, S., Han, C., Dong, Z. S., & Meng, L. (2019). A multi-stage stochastic programming model for relief distribution considering the state of road network. *Transportation Research Part B: Methodological*, *123*, 64 – 87.

Huang, M., Smilowitz, K. R., & Balcik, B. (2013). A continuous approximation approach for assessment routing in disaster relief. *Transportation Research Part B: Methodological*, *50*, 20 – 41.

Karlin, A. R., Manasse, M. S., Rudolph, L., & Sleator, D. D. (1988). Competitive snoopy caching. *Algorithmica*, *3*, 79—119.

Letchford, A. N., Nasiri, S. D., & Theis, D. O. (2013). Compact formulations of the steiner traveling salesman problem and related problems. *European Journal of Operational Research*, *228*, 83 – 92.

Li, Y., & Chung, S. H. (2019). Disaster relief routing under uncertainty: A robust optimization approach. *IISE Transactions*, *51*, 869–886.

Liao, C.-S., & Huang, Y. (2014). The covering canadian traveler problem. *Theoretical Computer Science*, *530*, 80—88.

Lu, C.-C., Ying, K.-C., & Chen, H.-J. (2016). Real-time relief distribution in the aftermath of disasters – a rolling horizon approach. *Transportation Research Part E: Logistics and Transportation Review*, *93*, 1—20.

Moreno, A., Alem, D., Ferreira, D., & Clark, A. (2018). An effective two-stage stochastic multi-trip location-transportation model with social concerns in relief supply chains. *European Journal of Operational Research*, *269*, 1050 – 1071.

Ozkapici, D. B., Ertem, M. A., & Aygunes, H. (2016). Intermodal humanitarian logistics model based on maritime transportation in istanbul. *Natural Hazards*, *83*, 345—364.

Papadimitriou, C., & Yannakakis, M. (1991). Shortest paths without a map. *Theoretical Computer Science*, *84*, 127—150.

Poteyeva, M., Denver, M., Barsky, L. E., & Aguirre, B. E. (2007). Search and rescue activities in disasters. In *Handbook of Disaster Research* (pp. 200–216). New York, NY: Springer New York.

Rauchecker, G., & Schryen, G. (2019). An exact branch-and-price algorithm for scheduling rescue units during disaster response. *European Journal of Operational Research*, *272*, 352–363.

Rawls, C. G., & Turnquist, M. A. (2010). Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B: Methodological*, *44*, 521 – 534.

Rolland, E., Patterson, R. A., Ward, K., & Dodin, B. (2010). Decision support for disaster management. *Operations Management Research*, *3*, 68—79.

Safaei, A. S., Mohammad, S. F., & Paydar, M. (2018). Robust bi-level optimization of relief logistics operations. *Applied Mathematical Modelling*, *39*, 359—380.

Schryen, G., Rauchecker, G., & Comes, T. (2015). Resource planning in disaster response: Decision support models and methodologies. *Business and Information Systems Engineering*, *57*, 243—259.

Sheu, J.-B. (2007). Challenges of emergency logistics management. *Transportation Research Part E Logistics and Transportation Review*, *43*, 655 – 659.

Shiri, D., & Salman, F. S. (2017). On the online multi-agent o-d k-canadian traveler problem. *Journal of Combinatorial Optimization*, *34*, 453–461.

Shiri, D., & Salman, F. S. (2019a). Competitive analysis of randomized online strategies for the online multi-agent k-canadian traveler problem. *Journal of Combinatorial Optimization*, *37*, 848–865.

Shiri, D., & Salman, F. S. (2019b). Online optimization of first-responder routes in disaster response logistics. *IBM Journal of Research and Development*, *64*, 1–9.

Sleator, D., & Tarjan, R. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM*, *28*, 202—208.

Statheropoulos, M., Agapiou, A., Pallis, G. C., Mikedi, K., Karma, S., Vamvakari, J., Dandoulaki, M., Andritsos, F., & Thomas, C. L. P. (2015). Factors that affect rescue time in urban search and rescue (usar) operations. *Natural Hazards*, *75*, 57 – 69.

Tang, J., Zhu, K., Guo, H., Gong, C., Liao, C., & Zhang, S. (2018). Using auction-based task allocation scheme for simulation optimization of search and rescue in disaster relief. *Simulation Modelling Practice and Theory*, *82*, 132 – 146.

Tzeng, G.-H., Cheng, H.-J., & Huang, T. D. (2007). Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E: Logistics and Transportation Review*, *43*, 673—686.

Wang, Q., & Nie, X. (2019). A stochastic programming model for emergency supply planning considering traffic congestion. *IISE Transactions*, *51*, 910–920.

Wenger, D. E. (1990). *Volunteer and organizational search and rescue activities following the Loma Prieta Earthquake: An integrated emergency and sociological analysis*. Technical Report College Station, TX: Texas A&M University, Hazard Reduction and Recovery Center.

Wex, F., Schryen, G., Feuerriegel, S., & Neumann, D. (2014). Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, *235*, 697—708.

Wex, F., Schryen, G., & Neumann, D. (2013). Assignments of collaborative rescue units during emergency response. *International Journal of Information Systems for Crisis Response and Management*, *5*, 63—80.

Yan, S. Y., & Shih, Y.-L. (2009). Optimal scheduling of emergency roadway repair and subsequent relief distribution. *Computers and Operations Research*, *36*, 2049—2065.

Zhang, H., Tong, W., Lin, G., & Xu, Y. (2019). Online minimum latency problem with edge uncertainty. *European Journal of Operational Research*, *273*, 418–429.

Zhang, H., Tong, W., Xu, Y., & Lin, G. (2015). The steiner traveling salesman problem with online edge blockages. *European Journal of Operational Research*, *243*, 30—40.

Zhang, H., Tong, W., Xu, Y., & Lin, G. (2016). The steiner traveling salesman problem with online advanced edge blockages. *Computers and Operations Research*, *70*, 26—38.

Zhang, H., Xu, Y., & Qin, L. (2013). The k-canadian travelers problem with communication. *Journal of Combinatorial Optimization*, *26*, 251—265.

Zhang, J.-H., Li, J., & Liu, Z.-P. (2012). Multiple-resource and multiple-depot emergency response problem considering secondary disasters. *Expert Systems with Applications*, *39*, 11066—11071.