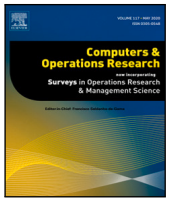
Computers & Operations Research 159 (2023) 106314

Contents lists available at ScienceDirect

Computers and Operations Research

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



Online optimisation for ambulance routing in disaster response with partial or no information on victim conditions

Davood Shiri [a], Vahid Akbari [b,*], Hakan Tozan [c]

[a] Sheffield University Management School, University of Sheffield, Conduit Road, Sheffield, S10 1FL, United Kingdom
[b] Nottingham University Business School, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom
[c] College of Engineering and Technology, American University of the Middle East, Kuwait

ARTICLE INFO

Keywords:
Ambulance routing
Mass emergency incident
Online optimisation
Competitive ratio
Partial information
Disaster relief

ABSTRACT

In response to mass casualty incidents, medical aid must be provided to numerous victims synchronously under challenging circumstances including uncertainty about the condition of victims. Therefore, it is essential to have decision support tools which can generate fast solutions under uncertainty and utilise the available medical resources efficiently to provide victims with the needed treatments. We introduce an online optimisation problem for routing and scheduling of the ambulances under uncertainty about the triage levels and required treatment times of the victims in mass casualty incidents. Due to the lack of information in the initial emergency response phase, we assume that the triage level and treatment time of each victim can be disclosed online only once the condition of a victim is closely assessed by the medical team on one of the ambulances at the casualty location. We investigate this problem under two different scenarios with partial and no information about the conditions of victims. We follow the theoretical competitive analysis framework for online optimisation and prove the lower bounds on the competitive ratio of deterministic and randomised online solutions for both cases of partial and no prior information. Next, we introduce three novel online heuristics to solve this problem. We verify the quality of our online solutions against the offline optimal solutions that are provided under complete information on a comprehensive set of 1296 instances from the literature. Finally, we draw our conclusions in regard to the suitability of each of our solutions in various scenarios of information availability with different numbers of victims.

1. Introduction

The number of mass casualty incidents either caused by natural (e.g., earthquakes) or human-made disasters (e.g., terrorist attacks) has been increased significantly worldwide in recent years (CRED & UNDRR, 2020). These incidents require urgent response from medical personnel to provide medical aid to several victims on the field under extremely challenging scenarios including uncertainty about the condition of victims needing help (Farahani et al., 2020; Tippong et al., 2022). Hence, it is crucial to have a decision-support tool which efficiently utilises the medical resources in such incidents. In this work, we focus on transportation of paramedics by ambulances in the immediate aftermath of mass casualty incidents. In such chaotic scenarios with numerous victims and limited resources, although the road conditions (e.g., information about damaged transportation links) and the location of victims can be learned by means of technologies such as satellites or drones or reports by rescuers who are at victim locations (Moreno et al., 2019; Kasaei and Salman, 2016), the *triage* (i.e., degree of severity of injury) of the victims cannot be classified remotely and must be determined by paramedics after medical examination of the victims on the field.

We add the uncertainty about the triage of victims to modelling and decision-making procedure for the *Ambulance Routing Problem in disaster response* (ARP-DS) which was first introduced by Talarico et al. (2015). Since each mass casualty incident is unique and unpredictable, past data is not relevant for this type of incidents in the majority of scenarios (Akbari and Shiri, 2022). Accordingly, the incomplete information cannot be modelled or approximated in a probabilistic way. In the context of the ARP-DS, applying a stochastic optimisation approach can lead to inaccurate or even infeasible solutions due to absence or inaccuracy of relevant past data (Jaillet and Wagner, 2008; Epstein, 2009; Xu and Gautam, 2020; Dwibedy and Mohanty, 2022). Similarly, robust optimisation approaches which are based on the worst-case perspective are not efficient due to lack of sufficient historical information (Jaillet and Stafford, 2001; Jaillet and Wagner, 2008). Therefore, the need for

* Corresponding author.
E-mail addresses: d.shiri@sheffield.ac.uk (D. Shiri), vahid.akbari@nottingham.ac.uk (V. Akbari), hakan.tozan@aum.edu.kw (H. Tozan).

https://doi.org/10.1016/j.cor.2023.106314
Received 7 September 2022; Received in revised form 26 May 2023; Accepted 14 June 2023
Available online 17 June 2023

online optimisation algorithms that can learn the uncertain information piece-by-piece and update their decisions dynamically to effectively utilise the revealed information and to efficiently dispatch ambulances in real-time is highly significant.

While the ARP-DS is a well-studied problem in the literature (Talarico et al., 2015; Tlili et al., 2017; Talebi et al., 2021; Aringhieri et al., 2022), to the best of our knowledge, none of the previous works have yet analysed this problem in the online optimisation framework. We extend the ARP-DS, to an online variation where the triage of all or some of the victims are not known to the paramedics (ambulances) a priori. Instead, for a victim whose triage is not known, the triage is determined online once that victim is examined by the paramedics at the victim location. We refer to this problem as the *Ambulance Routing Problem in disaster response with online triage* (ARP-DSOT) hereafter. Note that in the ARP-DSOT, once the triage of a victim is learned, the service time for treatment of that victim is also determined online as the victim conditions are examined by the paramedics.

We tackle the ARP-DSOT from both theoretical and practical viewpoints. We first follow the theoretical *competitive analysis* approach to formally analyse the problem from a worst-case perspective. For that, we prove a lower bound on the competitive performance of deterministic online algorithms for the case with no information. We then apply the notion of *randomisation* (Ben-David et al., 1994; Borodin and El-Yaniv, 2005; Hertz et al., 2018; Ashlagi et al., 2022) to further investigate the problem theoretically from a worst-case perspective by proving a lower bound on the expected competitive ratio of randomised solutions for the case with no information. Furthermore, we analyse the case with partial information and propose two lower bounds on the competitive ratio and the expected competitive ratio of deterministic and randomised algorithms, respectively. We will show that these lower bounds are functions of the ratio of partial known information. We also show that all of our lower bounds (for both cases of partial and no information) are increasing in the same order as the number of victims. This result implies the complexity of the online problem and confirms that under a worst-case scenario, no online algorithm can produce high quality solutions for the ARP-DSOT against an offline optimal solution.

Next, we apply various heuristic rules and design three novel online algorithms to tackle the ARP-DSOT from a computational perspective. One of our algorithms is a hybrid approach based on a combination of solving a mathematical formulation associated with the offline problem and a number of heuristic policies. While the CPU run time of this optimisation-based heuristic grows significantly with increasing the number of victims, our other two algorithms are purely based on heuristic policies and hence their computational running times are low, i.e., less than 1 s even for the largest benchmark instances provided in Talarico et al. (2015). Moreover, in our computational experiments, we show that the optimisation-based algorithm does not necessarily yield better solutions compared to the other two. Hence, these two algorithms are perfectly suitable to be applied in real-life post-disaster situations as they can produce and update solutions with respect to the new information in a very short time. We verify the quality of our three algorithms against the exact offline solutions which are optimally solved with access to prior information about the triage of all victims on an extensive set of instances from the literature.

The remainder of this article is organised as follows. We review the related literature in Section 2. We provide the detailed explanations of the ARP-DS and the ARP-DSOT in Section 3. In Section 4, we first give preliminary definitions and specify formal measure of performance for online optimisation and then we conduct a comprehensive theoretical worst-case competitive analysis on the ARP-DSOT. We introduce our online heuristic algorithms to solve the ARP-DSOT in Section 5. Our computational experiments which verify the quality of our online algorithms are presented in Section 6. Lastly, concluding remarks and future research directions are given in Section 7.

2. Literature review and scientific contributions

In this section, we first provide the related offline optimisation studies that address ambulance routing problem with access to full information and then review the related online optimisation literature.

2.1. Related offline problems

Among the articles which study the offline vehicle routing problems in a post-disaster situation, we present the related literature with a particular focus on fleet management for the ambulances. That is, we present the review of articles which consider ambulance routing while incorporating hospital capacities and victim groups in mass-casualty incidents. The offline version of our problem (i.e., the ARP-DS) is studied first in Talarico et al. (2015), where the problem of identifying routes of ambulances for serving a given set of red and green code victims is investigated under the assumption of having complete input information. In their problem, the objective is to minimise a weighted summation of the latest completion times of red and green victims. Talarico et al. (2015) presented two exact mathematical formulations to solve the ARP-DS. They also proposed an efficient large neighbourhood search (LNS) heuristic procedure to solve larger instances of this problem. Moreover, they introduced a comprehensive set of 1296 ARP-DS instances and tested their mathematical formulations and heuristic procedure on them.

Tlili et al. (2017) modelled the offline ARP-DS as an open vehicle routing problem where the objective is to minimise the total travel distance. They presented a cluster-first route-second solution based on the Petal algorithm and the particle swarm optimisation approach. Tlili et al. (2018) further investigated the ARP-DS by providing an efficient genetic based algorithm and testing it on small case studies. In a recent study, Aringhieri et al. (2022) investigated a variant of the ARP-DS by considering fairness and equity to provide services to victims. They modelled the problem as a new variation of the team orienteering problem and proposed a novel hybrid solution which is based on a machine learning and neighbourhood search. Talebi et al. (2021) applied the Multi-Objective Bees algorithm to tackle a variation of the ARP-DS where two groups of red and green victims are considered and the objective is to minimise the total travel distance and the latest service completion time simultaneously.

Other variations of the ambulance routing problem have also been addressed in the literature in recent years. Salman and Gül (2014) omitted triage and proposed a mixed integer programming model that simultaneously optimises capacity allocation and casualty transportation decisions. Their model minimised a weighted sum of the total travel and waiting time of casualties over the search-and-rescue period as well as the total cost of establishing new facilities. Zidi et al. (2019) also studied a version of the ambulance routing problem by ignoring triage where the objective is to minimise the total travel time. They proposed a new approach which is based on the hybridisation of Simulated Annealing and Tabu Search by applying the cluster-first route-second method. Tikani and Setak (2019) tackled the problem under the assumption of having complete prior information about triage and survival function of victims. They presented a mathematical model to obtain route plans where the objective is minimising the latest service completion time among the victims. They also provided two meta-heuristic procedures based on genetic algorithm and tabu search to provide solutions for larger instances of the problem. In another study, Rabbani et al. (2021) tackled a variant of the problem in Tikani and Setak (2019) by providing a mixed integer programming formulation as well as a Non-dominated Sorting Genetic Algorithm.

A remotely related stream of research to the offline ARP-DS is disaster relief routing, which includes a vast amount of articles. We refer the reader to surveys by De la Torre et al. (2012), Aringhieri et al. (2017) and Anuar et al. (2021) for this vein of work. All the articles discussed in this section differ from our study since they operate under complete input information. In our article, we address the ambulance routing problem by incorporating online uncertainty about the triage and treatment times of victims.

2.2. Related stochastic and online problems

We note that there is a stream of articles related to dynamic vehicle routing problem which are not defined within the context of ambulance routing, e.g., see the review articles of Ghiani et al. (2003), Pillac et al. (2013), Ojeda Rios et al. (2021) and Soeffker et al. (2022). These works differ from our problem since they do not involve all or some of the key features of the ARP-DSOT such as triage, transferring victims to hospitals, as well as hospital capacities. Since we study the transportation of ambulances while incorporating uncertainty about triage, the main focus of the literature review in this section is on routing of the ambulances in post-disaster situations while considering different forms of uncertainty.

A dynamic stochastic ambulance routing problem which does not involve triage is studied in Schilde et al. (2011) where victims must be transferred from their locations to the hospitals or back to their locations from the hospitals. In this problem, some of the requests are known a priori, while some of the requests are dynamic or associated with probability distributions. They modelled the problem as a dynamic stochastic Dial-a-Ride problem with expected return transports and proposed four different modifications of a meta-heuristic algorithm. Oksuz and Satoglu (2020) relied on known prior probabilistic information and developed a two-stage stochastic programming model to plan casualty transportation, accounting for triage, along with locating temporary medical centres and applied it on a case study. Yoon and Albert (2020) investigated the problem of dynamic ambulance routing problem under the uncertainty about the victim conditions by assuming prior probabilistic knowledge about the uncertain information. In their study, once the location of a victim is identified, the decision maker should decide which type of ambulance should be sent to the location. They analysed the problem by assuming that the condition of each victim follows a known probability distribution and provided a Markov decision process formulation that dynamically determines which to dispatch which vehicles to which victims. Recently, Lee et al. (2022) analysed the problem of relocation and routing of ambulances by assuming a known probability distribution for demand where the objective is to minimise the total time to deliver victims to hospitals. They applied a truncated Poisson distribution for forecasting future demands and adapted a branch-and-bound based Lagrangian dual decomposition to solve the problem.

Here, we note that the used methodologies in all the aforementioned articles significantly rely on the assumed probability distributions and hence differ from our online optimisation approach which is suitable for post-disaster scenarios where collecting probability distributions is not feasible. In our work, we investigate the ambulance routing problem under uncertainty within the framework of competitive analysis and online optimisation. In this approach, we do not consider any prior probabilistic knowledge associated with the uncertain information.

In recent years, vehicle routing problems with *online uncertainty* have been extensively investigated in the literature, e.g., Jaillet and Lu (2014), Büttner and Krumke (2016), Shiri and Salman (2020), Akbari and Shiri (2021) and Zhang et al. (2022). In the online vehicle routing problem literature, a remotely relevant problem to our study is the online multi-server Dial-a-Ride problem in which requests are not known a priori and are disclosed over time (Bonifaci et al., 2006; Luo and Schonfeld, 2011). However, this problem differs from the ARP-DSOT as it does not involve triage, hospital capacities, and ambulance routing. To the best of our knowledge, none of the articles in the literature of online optimisation have investigated the routing and scheduling of ambulances so far. Our work is a first attempt to model triage in the form of online uncertainty for this problem in the context of mass casualty incidents.

3. Offline and online problem definitions

In this section, we first give the formal description of the offline problem (i.e., ARP-DS) that was introduced in Talarico et al. (2015) in which all the input parameters are known in advance. We then give the detailed description of the online version (ARP-DSOT) together with the online parameters and specify how their exact values can be obtained.

3.1. The offline problem

Since the aim of our study is to incorporate the online triage uncertainty to the problem (i.e., we do not intend to provide policies for the offline version), we respect the conventional and standard problem definition of the offline problem given in the literature. The existing standard literature of post-disaster ambulance routing problems (Talarico et al., 2015; Tlili et al., 2017; Aringhieri et al., 2022) considers two triage levels for the victims in the context of the ARP-DS; **red code**: a seriously injured person who must be transferred to a hospital by an ambulance, and **green code**: a slightly injured person who can be treated directly in the field by the paramedics on the ambulances. We refer the reader to Talarico et al. (2015) for detailed motivations of using two groups of green and red victims to distinguish the fundamental operations of ambulances in disaster response scenarios.

As discussed, the ARP-DS is introduced under the assumption of having complete information regarding the triage and the required treatment time of each victim. That is, the ambulances have complete information about the red and green code victims before they start their routes in the offline problem. In the ARP-DS, each ambulance can carry one red code victim at a time and each red code victim is immediately delivered to a hospital after being picked up. Because green code victims can be treated at their locations, an ambulance can go directly to another victim location after servicing a green code victim. Hence, an ambulance can service multiple victims on its route before returning to a hospital. Here, for clarity, a route is referred to as a tour that starts at a hospital, visits one or more victims, and ends at a hospital. Therefore, in a solution of the ARP-DS, an ambulance may perform multiple routes.

Here, we can formally present the definition of the offline ARP-DS as follows. We represent the set of victims $V = R \cup G$, where R and G are sets of red and green code victims, respectively. We let H represent the set of hospitals and C_h is the capacity of hospital $h \in H$ such that $\sum_{h\in H} C_h \geq |R|$, so that each red victim can be allocated to a hospital. We denote the set of ambulances by A (such that $|A| \leq |V|$) and the initial location of ambulance $a \in A$ by $d_a \in H$. Therefore, we define the set of nodes by $N = R \cup G \cup H = V \cup H$ and the set of directed links by $E = \{V \times V\} \cup \{V \times H\} \cup \{H \times V\}$. We denote the travel time of link $(i, j) \in E$ by t_{ij}. Each victim $v \in V$ is associated with a treatment time S_v and each hospital $h \in H$ is associated with a service time S_h to drop off a red code victim at hospital h.

In the ARP-DS, weights W_R and W_G are associated with red and green code victims, respectively. These weights reflect real-world triage levels for a mass-casualty incident in order to assign a higher priority to victims whose treatments are more urgent. Accordingly, the offline ARP-DS is to identify the routes of the ambulances for servicing red and green code victims such that the weighted summation of the latest service completion time among the red code victims and the latest service completion time among the green code victims is minimised. That is, the objective function minimises the weighted summation of the longest waiting time over all of the victims in each of the triage levels. The used notation in the problem definition of the ARP-DS is presented in Table 1.

Table 1
Table of notation for the ARP-DS and the ARP-DSOT.

Notation	Description
R^1	Set of known red code victims
G^1	Set of known green code victims
R^2	Set of unknown red code victims **(unknown in the ARP-DSOT)**
G^2	Set of unknown green code victims **(unknown in the ARP-DSOT)**
$R = R^1 \cup R^2$	Set of all red code victims **(unknown in the ARP-DSOT)**
$G = G^1 \cup G^2$	Set of all green code victims **(unknown in the ARP-DSOT)**
$V = R \cup G$	Set of victims
H	Set of hospitals
C_h	Capacity of hospital $h \in H$
A such that ($\|A\| \leq \|V\|$)	Set of ambulances
A_h	Set of ambulances that are initially located at hospital $h \in H$
f_h^a	Equals 1 if ambulance a is initially located in hospital h, 0 o/w.
$d_a \subset H$	Initial location of ambulance $a \in A$
$E = \{V \times V\} \cup \{V \times H\} \cup \{H \times V\}$	Set of directed links
t_{ij}	Travel time of link $(i, j) \in E$
S_v	Treatment time of victim $v \in V$ **(unknown in the ARP-DSOT)**
S_h	Service time to drop off a red code victim at hospital $h \in H$
W_R	Weight of red code victims
W_G	Weight of green code victims

Table 2
Decision variables used in the mathematical models.

Notation	Type	Description
x_{ij}	$\{0, 1\}$	Equals 1 if an ambulance serves victim i directly before victim j.
u_{vh}	$\{0, 1\}$	Equals 1 if victim $v \in V$ is brought to hospital $h \in H$.
b_v	≥ 0	Visiting time of victim v
e_R	≥ 0	Last service completion time among red victims.
e_G	≥ 0	Last service completion time among green victims.

3.1.1. Mathematical formulation for the offline problem

The offline problem described above was studied first in Talarico et al. (2015) where the case with full information about the treatment times and triage levels was analysed and an exact mathematical model to tackle this problem was developed. We utilise the exact formulation of the offline problem for devising our hybrid online algorithm as well as computing the experimental competitive ratios of our algorithms. In the following, we first give the decision variables in Table 2 and then present the formulation.

$$\min \quad W_R \cdot e_R + W_G \cdot e_G \tag{1}$$

$$\sum_{j \in V \cup H} x_{hj} \leq |A_h| \quad \forall h \in H \tag{2}$$

$$\sum_{j \in V \cup H} x_{ji} = \sum_{j \in V \cup H} x_{ij} = 1 \quad \forall i \in V \tag{3}$$

$$b_v + S_v + t_{vj} \leq b_j + (1 - x_{vj}) \cdot M \quad \forall v \in G \cup H; \forall j \in V \tag{4}$$

$$b_v + S_v + t_{vh} + S_h + t_{hj} \leq b_j + (2 - x_{vj} - u_{vh}) \cdot M \quad \forall v \in R; j \in V; h \in H \tag{5}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \tag{6}$$

$$\sum_{h \in H} u_{vh} = 1 \quad \forall v \in R \tag{7}$$

$$\sum_{v \in R} u_{vh} \leq C_h \quad \forall h \in H \tag{8}$$

$$e_G \geq b_v + S_v \quad \forall v \in G \tag{9}$$

$$e_R \geq b_v + S_v + u_{vh} \cdot (t_{vh} + S_h) \quad \forall v \in R; h \in H \tag{10}$$

$$b_v \geq 0 \quad \forall v \in V \cup H \tag{11}$$

$$u_{vh} \in \{0, 1\} \quad \forall v \in R; h \in H \tag{12}$$

Here (1) defines the objective function. Constraints (2), restricts the number of ambulances that start from each hospital. Constraint set (3) is to ensure the flow balance and connectivity of the ambulance routes. Constraints (4) and (5) calculate the time progression of green and red victims. By constraints (4), if a green or red victim $j \in V$ is immediately visited after a green victim $v \in G$ by the same ambulance, the time at which the ambulance reaches to victim j is equal to the time when the ambulance reaches to the green victim v (b_v), plus the time the medical personnel provided the service to this victim (S_v) and the time that the ambulance travelled the distance from the location of victim v to the location of victim j (t_{vj}). Constraints (5) follow a similar procedure to calculate the visiting time of a red or green victim $j \in V$ (b_v) who is visited immediately after a red victim $v \in R$ by the same ambulance. In this case, the visiting time of victim $v \in R$ (b_v), service time of victim v at their location (S_v), the time to take victim v to a hospital (t_{vh}), the service time at the hospital (S_h) and the time to travel from the hospital to the location of victim j (t_{hj}) should be considered in the time calculations. In this case however, the time to travel from victim location $v \in R$ to victim location $j \in V$ should not be considered as the ambulance does not travel directly from location of victim v to j. In both of these constraints, selecting any M such that:

$$M \geq \sum_{v \in V} S_v + |R| \max_{h \in H} S_h + \sum_{g \in G} \max_{j \in V \neq g} t_{gj} + \sum_{r \in R} \max_{h \in H} t_{rh} + \sum_{h \in H} |A_h| \max_{v \in V} t_{hv}$$

guarantees a sufficiently large value for M. This term includes serving time of all victims, highest service time at hospitals considered for all red victims, highest possible travelling time from each green victim to other victim locations, highest possible travelling time from each red victim to a hospital and highest possible travelling time from each hospital to a victim location for the ambulances that are initially positioned in those hospitals. Constraints (6) define the 2-index x_{ij} variables. Constraints (7) ensure that all the victims with a red triage level are transferred to a hospital. Constraints (8) enforce the capacity restrictions in the hospitals. Constraints (9) and (10) calculate the last time a green and a red victim are served, respectively. The rest of the constraints are to set the variable domains.

3.2. The online problem

In this section, we present the online problem (i.e., ARP-DSOT) and state the differences between the parameters of this problem and the offline version. While in the ARP-DSOT, some of the parameters are not known in advance, the road conditions (i.e., E and t_{ij} for

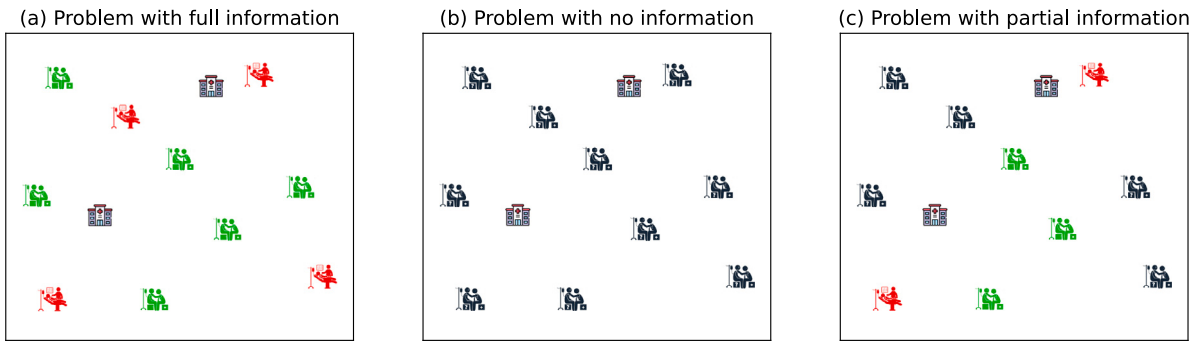


Fig. 1. An illustration of the problem under various scenarios for information availability.

$(i, j) \in E$), set of victims (i.e., V), and their locations are known to the ambulances. This is because this information can be obtained by means of technologies (e.g., drones and satellites) or transmitted reports by people or the rescue teams (Talarico et al., 2015; Kasaei and Salman, 2016; Akbari et al., 2021).

However, the triage levels (sets R and G) and the required treatment times (S_v for $v \in V$) are not known before the paramedics assess the condition of the victims on the scene. With regards to the online optimisation literature (Jaillet and Wagner, 2006; Chen and Nie, 2015), we distinguish between three cases of information availability to model different real-world scenarios; **full information:** the ambulances have complete information about triage and treatment times of all victims, **partial information:** the ambulances have complete information about triage and treatment times for only a subset of victims and have no information about triage and treatment times of the other victims, and **no information:** the ambulances have no information about triage and treatment times of any victims. Fig. 1 illustrates the problem under different scenarios of information availability. In this figure, a victim coloured by red, represents a red victim whose triage is known. Similarly, a victim coloured by green, denotes a green victim with a known tirage. However, a victim coloured by dark grey, corresponds to a victim whose triage is unknown and can only be revealed after the victim is examined by the paramedics in one of the ambulances on the scene. While the primary focus of our study is to address the case with no information, we present our problem definition and online algorithms such that they cover the case with partial information as well. This enables us to conduct a comprehensive sensitivity analysis by considering scenarios with available partial information in our computational experiments.

To cover all the three cases mentioned above, we assume that the triage and treatment times for a subset of victims (i.e., set $R^1 \cup G^1$) can be specified a priori (e.g., by the rescue teams who are on the field), whereas the triage and treatment times for the remaining subset of victims (i.e., set $R^2 \cup G^2$) are not known to the ambulances beforehand (e.g., the locations of these victims are reported by inexpert people or technological tools such as drones with limited access to the victims) and can only be revealed online once these victims are closely examined at their locations. We remark that when $V = R^1 \cup G^1$ and $V = R^2 \cup G^2$ the problem corresponds to the cases with full and no information, respectively. Also, when both $R^1 \cup G^1$ and $R^2 \cup G^2$ are non-empty, the problem corresponds to the case with partial information.

We note that when $R^2 \cup G^2 \neq \emptyset$ (i.e., the cases with partial or no information), it is not possible to provide a static routing plan for the ambulances and the routing decisions should be updated dynamically as new information reveals about the victims in $R^2 \cup G^2$. Formally, for a victim $v \in R^2 \cup G^2$, the triage level (i.e., if $v \in G$ or $v \in R$) and the treatment time (i.e., S_v) are learned online when that victim is examined by the paramedics on one of the ambulances. Once this information is revealed, it is immediately shared with the other paramedics in rest of the ambulances. We assume that the categorisation time for identifying the triage level of the victims is negligible compared to the travel and treatment times.

Other parameters of the ARP-DSOT are known and exactly the same as those of the ARP-DS. The notations for the pieces of information that are not known in the ARP-DSOT are denoted in Table 1. These information highlight the differences in the problem inputs of the ARP-DS and the ARP-DSOT. To be consistent with the literature of the ARP-DS and to be able to compare our online solutions with the solutions of the same instances in the offline version, in the ARP-DSOT we utilise the same objective function as was presented for the ARP-DS. In the ARP-DSOT, decisions regarding prioritisation of the victims to be treated should be made online each time a new piece of information is revealed. For example, if an ambulance identifies a green victim, it should be decided whether the ambulance should service the green code victim immediately or the ambulance should pass the green victim without providing service in order to prioritise servicing potential red code victims. In addition, decisions regarding choosing the hospital to which a red victim should be taken must be made online with respect to the capacity of the hospitals at the moment the red victim is identified.

4. Worst-case competitive analysis of the ARP-DSOT

Online optimisation problems are analysed within the theoretical framework of the so called *competitive analysis* in the literature (see Jaillet and Wagner, 2008; Xu and Gautam, 2020; Wang et al., 2022; He et al., 2022; Hertrich et al., 2022 for examples). In this framework, the performance of the solution obtained under incomplete information, **online solution**, is compared with the performance of the optimal solution obtained in presence of complete information, **offline solution**. The competitive analysis approach which was first introduced by Sleator and Tarjan (1985) is based on identifying the worst-case performance of online solutions and hence no probabilistic information is required in this approach. The goal in online optimisation is to generate online solutions which guarantee a performance as close as possible to the offline optimal solutions. The *competitive ratio* analyses the proximity of an online solution to the optimal offline solution (Liu et al., 2005; Jaillet and Wagner, 2006; Zhang et al., 2016; Chen et al., 2020). Formally, the competitive ratio of a deterministic online solution (ALG_D) applied to an online minimisation problem is the supremum of the ratio of the objective function value of the online solution to the objective function value of the offline optimal solution (OPT) over all instances ($\mathcal{I}$) of the problem, i.e.,

$$\sup_{\delta \in \mathcal{I}} \frac{obj(ALG_D(\delta))}{obj(OPT(\delta))}.$$

With the aim of improving the *expected* competitive performance of online algorithms, the notion of randomisation has been also utilised in the literature (Ben-David et al., 1994; Azar and Epstein, 1998; Borodin and El-Yaniv, 2005; Ashlagi et al., 2022; Shiri and Tozan, 2022). In this approach, an online algorithm relies on random choices based on probability distributions to make decisions, i.e., a randomised online

solution may output a different solution every time that it is applied to the same problem instance. For a randomised online solution (ALG_R) and a minimisation problem, the *expected competitive ratio* is defined and can be computed by finding the supremum of the ratio of the expected objective function value of the online solution to the objective function value of the offline optimal solution (OPT) over all problem instances ($\mathcal{I}$), i.e.,

$$\sup_{\delta\in\mathcal{I}} \frac{\mathbb{E}[obj(ALG_R(\delta))]}{obj(OPT(\delta))}.$$

Here, we remark that while randomisation can result in improved theoretical expected competitive ratios, in practise, developing randomised solutions might result in very poor solutions and hence taking random actions upon discovery of new information is flawed. As a result of this, to provide a more detailed theoretical analysis of the problem, we only investigate randomised solutions theoretically and do not incorporate randomisation in our practical solution algorithms.

Recently, to compare the performance of online algorithms on real-world instances, the notion of *experimental competitive ratio* have been used extensively (Legrain and Jaillet, 2016; Akbari et al., 2021; Wang et al., 2022; Zhang et al., 2022). Formally, for a given set of instances Δ, the experimental competitive ratio of an online algorithm (ALG) is the average ratio of the objective function value of the online algorithm over the objective function value of the offline optimum (OPT) on those instances, i.e.,

$$\frac{\sum_{\delta\in\Delta} obj(ALG(\delta))}{\sum_{\delta\in\Delta} obj(OPT(\delta))}.$$

4.1. Lower bounds for the case with no information

We first prove that increasing the number of ambulances does not improve the competitive ratio in a worst-case instance of the problem.

Lemma 4.1. *The competitive ratio of an optimal online algorithm for the ARP-DSOT is non-decreasing in $|A|$.*

Proof. Consider an instance where there is an ambulance $a_1 \in A$ positioned at hospital $h_1 \in H$ (whose capacity is $C_{h_1} = |V|$) and other ambulances in $A\setminus\{a_1\}$ are positioned at hospitals in $H\setminus\{h_1\}$. Suppose that M is a sufficiently large positive number. We set $t_{ij} = M$ such that the link (i, j) is blocked and untraversable for the cases where: (1) both $i \in V$ and $j \in V$ are victim nodes, (2) both $i \in H$ and $j \in H$ are hospital nodes, (3) $i \in H\setminus\{h_1\}$ and $j \in V$, (4) or $i \in V$ and $j \in H\setminus\{h_1\}$, i.e., these blocked links can be justified as damaged transportation roads by disaster in an artificial worst-case scenario (Shiri et al., 2020; Sayarshad et al., 2020). Therefore, a worst-case problem instance for the case where $|A| = 1$ can be simulated based on the above setting for any $|A| > 1$. □

In the following, we derive a lower bound of $|V|$ on the competitive ratio of deterministic online solutions for the ARP-DSOT where $|V|$ denotes the number of victims. This lower bound means that the objective function value of an optimal deterministic online solution would be at least $|V|$ times of the objective function value of the offline optimal solution in a worst-case problem instance. For proving the lower bound, we specify the characteristics of an ARP-DSOT instance which imposes the competitive ratio of $|V|$ on any deterministic online solution.

Lemma 4.2. *No deterministic online algorithm achieves a competitive ratio less than $|V|$ for the ARP-DSOT.*

Proof. According to Lemma 4.1, we analyse an instance with one hospital $h_1 \in H$ with capacity $C_{h_1} = |V| = n$ and one ambulance $a_1 \in A$. Since we consider the case with no information, we set $R^1 \cup G^1 = \emptyset$ and $V = R^2 \cup G^2 = \{v_1, v_2, \dots, v_n\}$ (i.e., $|V| = n$). Suppose that M and ϵ are sufficiently large and small positive numbers, respectively, such that $M \times \epsilon$ approaches 0. We set $t_{ij} = 1$ for the case where i or j is a victim node and the other one is hospital h_1. We also set $S_v = \epsilon$ for $v \in V$, $S_{h_1} = \epsilon$ for h_1, $W_R = M$, $W_G = \epsilon$. The topology of this worst-case instance has been depicted in Fig. 2. This graph is a star tree with node set $\{h_1, v_1, v_2, \dots, v_{n-1}, v_n\}$ where h_1 is the root node and the travel time between hospital h_1 and victim nodes $(\{\{h_1\}\times V\}\cup\{V\times\{h_1\}\})$ is 1.

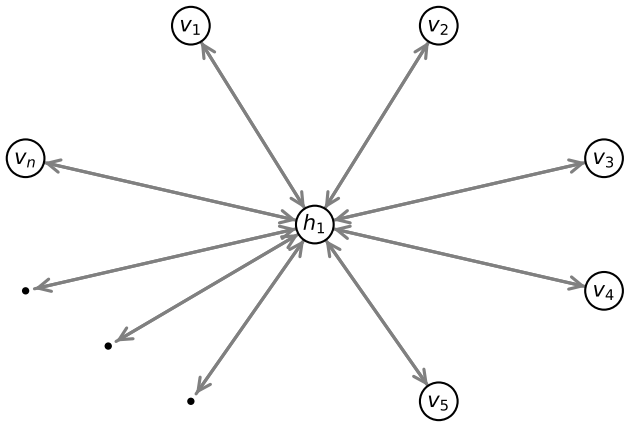


Fig. 2. The used instance for the proof of Lemma 4.2.

In the offline optimal solution of the instance described above, the information about the red and green victims and their treatment times are known a priori. Suppose that the only red victim is at $v_* \in V$. Since $W_R \gg W_G$ (i.e., $M \gg \epsilon$), the ambulance a_1 first traverses the arc (h_1, v_*) and services the only red victim $v_* \in R$ by incurring a time of $t_{h_1v_*} + S_{v_*} = 1 + \epsilon$. Then, a_1 returns to h_1 by incurring a time of $t_{v_*h_1} + S_{h_1} = 1 + \epsilon$ to deliver v_* to the hospital. Thus, a total time of $2+2\epsilon$ is incurred for treating $v_* \in R$. Hence, the weighted latest service completion time of victims in R would be

$$W_R \times (2 + 2\epsilon) = M \times (2 + 2\epsilon).$$

Similarly, a time of

$$t_{h_1v_i} + S_{v_i} + t_{v_ih_1} = 1 + \epsilon + 1 = 2 + \epsilon$$

must be incurred for treating each of the victims in $G = \{v_1, v_2, \dots, v_n\}\setminus\{v_*\}$ and returning to h_1 (a_1 must return to h_1 to be able to visit other nodes due to the topology of the graph in Fig. 2 unless the victim is the last green victim). Also, the required time for visiting and treating the last green victim is $1 + \epsilon$. Note that the route of the ambulance in the offline optimum terminates at the node of the last green victim. Therefore, the weighted latest service completion time of victims in G would be

$$W_G\times\big((2+2\epsilon)+(n-2)\times(2+\epsilon)+(1+\epsilon)\big) = \epsilon\times\big((2+2\epsilon)+(n-2)\times(2+\epsilon)+(1+\epsilon)\big),$$

and hence the objective function of the offline optimal solution would be

$$M \times (2 + 2\epsilon) + \epsilon \times \big((2 + 2\epsilon) + (n - 2) \times (2 + \epsilon) + (1 + \epsilon)\big).$$

We note that any deterministic online solution in which the information about red and green victims is not known, corresponds to an order of $|V| = n$ iterations such that at each iteration a_1 traverses an arc (h_1, v_i) to reach the victim at $v_i \in V$, then services this victim before returning to h_1 by traversing the arc (v_i, h_1) (i.e., at each iteration a_1 must return to h_1 due to the topology of graph in Fig. 2). For any deterministic online solution, we consider the instance in which v_* is treated at the **last** (nth) iteration. Therefore, the latest service completion time of victims in G would be at least

$$W_G \times \big((n - 2) \times (2 + \epsilon) + (1 + \epsilon)\big) = \epsilon \times \big((n - 2) \times (2 + \epsilon) + (1 + \epsilon)\big),$$

and the latest service completion time of victims in R would be at least

$$W_R \times \big((n-1)\times(2+\epsilon)+(2+2\epsilon)\big) = M \times \big((n-1)\times(2+\epsilon)+(2+2\epsilon)\big),$$

and thus the objective function of any deterministic online solution applied to the instance described above would be at least

$$\epsilon \times \big((n-2)\times(2+\epsilon)+(1+\epsilon)\big) + M \times \big((n-1)\times(2+\epsilon)+(2+2\epsilon)\big).$$

We assumed that M and ϵ are sufficiently large and small positive numbers, respectively, such that $M \times \epsilon$ approaches 0. The lower bound of $|V| = n$ follows on the competitive ratio of any deterministic online solution based on the obtained objective function values for the online and offline optimum solutions. □

In the following we investigate the effect of randomisation on the expected worst-case performance of online algorithms by utilising Yao's Principle (Yao, 1977), e.g., see Caragiannis et al. (2008) and Shiri and Salman (2019) for proofs deriving bounds on the expected competitive ratio using Yao's principle.

Lemma 4.3. *No randomised online algorithm achieves an expected competitive ratio less than* $\frac{|V|+1}{2}$.

Proof. We consider the analysed instance in the proof of Lemma 4.2. We choose one of the victim nodes v_* uniformly at random and assume that the red victim is at v_*. As discussed in the proof of Lemma 4.2, any deterministic online algorithm corresponds to $|V|$ iterations such that at each iteration the ambulance a_1 visits one of the nodes and comes back to the hospital h_1. With respect to the probability distribution described above, the red victim is found with probability $\frac{1}{|V|}$ at iteration i for $i = 1, 2, \ldots, |V|$.

If the red victim is found at iteration i, the objective function value of a deterministic online algorithm would be at least $W_R \times \big((i-1)\times(2+\epsilon)+(2+2\epsilon)\big) = M \times \big((i-1)\times(2+\epsilon)+(2+2\epsilon)\big)$. Therefore, the expected objective function value of any deterministic online algorithm would be at least

$$\frac{1}{|V|}\sum_{i=1}^{|V|} M \times ((i-1)\times(2+\epsilon)+(2+2\epsilon)),$$

which approaches to

$$\frac{2M}{|V|}\sum_{i=1}^{|V|} i = M \times (|V|+1).$$

Note that the objective function value of the offline optimum is $M \times (2+2\epsilon) + \epsilon \times \big((2+2\epsilon)+(n-2)\times(2+\epsilon)+(1+\epsilon)\big)$, i.e., the same as the objective function value of the offline optimum in the proof of Lemma 4.2. The proof is complete by Yao's principle (Yao, 1977). □

4.2. Lower bounds for the case with partial information

We first prove updated bounds with respect to the assumption of having partial information. We then analyse the effect of increasing the percentage of partial information on the worst-case competitive ratio. We represent $\alpha = \frac{|R^1 \cup G^1|}{|V|} = 1 - \frac{|R^2 \cup G^2|}{|V|}$ as the percentage of known partial information. We remark that when $\alpha = 0$, the problem corresponds to the case with no information and the lower bounds proved in Lemmas 4.2 and 4.3 must be considered.

Lemma 4.4. *Suppose that* $V \neq |R^2 \cup G^2|$ *and* $\alpha = 1 - \frac{|R^2 \cup G^2|}{|V|}$. *The competitive ratio of no deterministic online algorithm is less than* $(1-\alpha)\times|V|$ *and the expected competitive ratio of no randomised online algorithm is less than* $\frac{(1-\alpha)\times|V|}{2} + \frac{1}{2}$.

Proof. Note that Lemma 4.1 is valid for the case with partial information as well. Hence, we consider the same instance as presented in the proof of Lemma 4.2 with only one difference such that $R^1 \cup G^1 \neq \emptyset$ and $R^2 \cup G^2 = V \setminus (R^1 \cup G^1)$. To construct a worst-case instance, we set $R^1 = \emptyset$, i.e., the only red victim $v_* \notin R^1$. Note that $W_R = M$ and $W_G = \epsilon$. In the offline optimum, the victims in G^1 are serviced after v_* and hence the latest service completion time of red victim v_* is $M \times (2+2\epsilon)$.

Similar to the proofs of Lemmas 4.2 and 4.3, the latest service completion time of red victim v_* is not less than $M \times \big((|R^2 \cup G^2| - 1)\times(2+\epsilon)+(2+2\epsilon)\big)$ and the (expected) latest service completion time of v_* is not less than $M \times (|R^2 \cup G^2| + 1)$. Given that $M \times \epsilon \to 0$, the effect of latest service completion time of green victims can be omitted from calculations of competitive ratio for the sake of simplicity, i.e., this is a valid statement by definitions of competitive ratio and expected competitive ratio. The proof is complete. □

4.3. Analysis of the lower bounds

All the lower bounds proved in Sections 4.1 and 4.2 increase in order of the number of victims, $|V|$. As it can be interpreted, the lower bounds grow significantly when the number of victims increases even when the percentage of known partial information is high. For example, for the case with 60% of known partial information and 50 victims, the lower bounds on the competitive ratios are 20 and 10.5 for deterministic and randomised algorithms, respectively. We will discuss in the next sections that the worst-case competitive ratio which is outputted by our online algorithms on benchmark instances with 50 victims without having access to partial information is 3.16.

This trend worsens as the number of victims increases. This means for scenarios with higher number of victims, even with a high α (i.e., percentage of known partial information), an optimal online algorithm that meets these bounds performs poorly against the offline optimum from a worst-case perspective. That being the case, although these bounds are tight theoretical results, they are not informative and insightful for real-world mass casualty scenarios. One reason for this limitation is that the worst-case instance which imposes the high competitive ratio is an artificial instance which is unlikely to occur in real-world. Accordingly, the optimal online algorithm which meets these bounds must be designed to be robust against such artificial worst-case instances rather than real-world scenarios. An approach which is popularly applied in recent years (Zhang et al., 2019; Akbari and Shiri, 2021; Wang et al., 2022; Zhang et al., 2022) to remedy this limitation is to design online algorithms that achieve good experimental competitive ratios. This approach leads to online algorithms which are suitable to be utilised on problem instances with real-world characteristics.

5. Online algorithms

In this section, we propose three deterministic online algorithms to tackle the ARP-DSOT. We refer to our solution procedures as the *optimisation-based*, *clustering*, and *utility-based* algorithms and describe them in detail in the following three subsections.

5.1. Optimisation-based algorithm

The optimisation-based algorithm is a hybrid two-phase approach which is based on a combination of solving the exact mathematical formulation presented in Section 3.1.1 and some heuristic rules. In the first phase, we utilise the available partial information and solve the mathematical formulation on a transformation of the problem instance to obtain an assignment of victims to each ambulance as well as the order of servicing the assigned victims by the ambulances. For that, we use the available triage and service times for victims in $R^1 \cup G^1$. However, since we have no information about the triage and treatment times of victims in $R^2 \cup G^2 = V \setminus (R^1 \cup G^1)$, we give them a higher priority than the already known green victims (i.e., G^1) and a lower priority than the known red victims (i.e., R^1). To achieve this, we initially set the triage of victims in $V \setminus (R^1 \cup G^1)$ to green, i.e., this

enforces lower priorities to victims in $V \setminus (R^1 \cup G^1)$ compared to victims in R^1 since $W_G \leq W_R$. We also set the treatment time of these victims to 0, i.e., this gives relatively higher service priorities to victims in $V \setminus (R^1 \cup G^1)$ compared to victims in G^1 due to the latency objective function (Ajam et al., 2022). We remark that these initial values will be revealed dynamically by the ambulances on the field and decisions will be updated accordingly by our heuristic rules in an online manner in the second phase.

The second phase commences once the optimisation model is solved on the transformed instance of the problem and the assignment of victims to the ambulances as well as the order of visits are specified. We denote this assignment and order for ambulance a_λ by L_λ and O_λ, respectively. Our objective in the second phase of the optimisation-based algorithm is to schedule the routes of the ambulances in a way that they follow the order of victim visits obtained from the optimisation step. We need to note that, when producing the routes for the ambulances, some victims with unknown triage, $V \setminus (R^1 \cup G^1)$, which are initially assumed as green victims in the first phase, may be classified as red victims by the paramedics when they are examined at their locations in the second phase. To address this, whenever an ambulance encounters a red victim whose triage was assumed to be green initially, the route of the ambulance is modified such that the red victim is transferred to a hospital with available capacity for which the summation of (1) the travel time between the location of the current red victim and that hospital, and (2) the travel time between that hospital and the next victim to be visited by the ambulance, is minimised. In this way, we adhere with the plan produced by the optimisation model while ensuring the feasibility of our online solution.

As a further improvement to the solution obtained by the optimisation model, we let the ambulances which have completed servicing all the victims that are assigned to them by the optimisation model (i.e., the victims in L_λ), to service the remaining victims (i.e., the victims that are not serviced yet by the other ambulances) in a greedy manner (i.e., by servicing the closest unserviced victim to the location of the ambulance) until all the victims are serviced. In this way, we avoid having idle ambulances before all the victims are served and hence we improve the quality of our solution. A high-level pseudo-code of the optimisation-based algorithm, only addressing the major aspects of the algorithm, is presented in Algorithm 1.

Algorithm 1 Optimisation-based algorithm

1: **Input:** an instance of the problem ▷ see Table 1
2: **Step 1:**
a: set $R = R^1$
b: set $G = V \setminus R^1$
c: set $S_v = 0$ for all $v \in V \setminus (R^1 \cup G^1)$
d: solve the model in Section 3.1.1
e: find L_λ and O_λ for all $a_\lambda \in A$
3: **Step 2:**
a: set $V' = V$ ▷ define V' to keep track of victims who are not serviced yet
b: **dispatch** ambulance a_λ to service victims with respect to O_λ
c: **if** ambulance a_λ encounters a green victim v_g, **then:** ▷ case 1
d: service the green victim v_g
e: set $V' = V' \setminus \{v_g\}$
f: **end if**
g: **if** ambulance a_λ encounters a red victim v_r, **then:** ▷ case 2
h: $v_* =$ the next victim in L_λ to be serviced by a_λ after servicing v_r
i: $h_* = \text{argmin}_{h \in H}\, t_{v_r h} + S_h + t_{h v_*}$
j: deliver v_r to h_*
k: update $C_{h_*} = C_{h_*} - 1$
l: set $H = H \setminus \{h_*\}$ if $C_{h_*} = 0$
m: set $V' = V' \setminus \{v_r\}$
n: **end if**
4: **Step 3:**
a: **if** ambulance a_λ completes servicing all victims in L_λ, **then:**
b: $v_* =$ the closest victim to a_λ in V'
c: dispatch a_λ to v_* and service it based on one of the two cases in step 2
d: set $V' = V' \setminus \{v_*\}$
e: **end if**
f: **if** $V' = \emptyset$, **then:** ▷ if all victims are serviced
g: **terminate** ▷ termination point
h: **end if**

5.2. Clustering algorithm

Not only solving the mathematical formulation in Section 3.1.1 requires access to an optimisation solver, it can be time consuming when the size of the problem instance increases. Therefore, the optimisation-based algorithm may fail to produce timely solutions for the emergency mass casualty incident scenarios. Hence the need for online algorithms that are able to provide solutions in a short time is extremely important to handle these scenarios. With that motivation, we design and develop the clustering algorithm which is an alternative two-phase solution to the optimisation-based algorithm with a considerably lower running time.

Similar to the optimisation-based algorithm, to give higher priorities to victims with unknown triage compared to the known green victims, we initially set the triage of victims in $V \setminus (R^1 \cup G^1)$ to green and set the treatment time of these victims to 0. Given the nature of the latency objective and with a service time of 0, we prioritise victims with unknown triage to the known green victims. In the first phase of the clustering algorithm, the victims are assigned to the ambulances iteratively based on the travel times between locations as well as the initial values for treatment times as follows. First, the known red victims in R^1 are uniformly assigned to the ambulances. Next, the victims with unknown triage, $V \setminus (R^1 \cup G^1)$, are uniformly assigned to the ambulances. Finally, the known green victims in G^1 are uniformly assigned to the ambulances. As a result of this assignment, a cluster of victims is assigned to each ambulance such that not only the geographical conditions of the victims, but also their triage is incorporated in the clustering procedure. A detailed pseudo-code of this clustering procedure is presented in Appendix. For an ambulance $a_\lambda \in A$, we represent this cluster by L_λ. Once the clusters are determined, the second phase of the algorithm begins.

Contrary to the optimisation-based algorithm where the mathematical formulation specifies the order in which the victims should be visited by the ambulances, in the clustering algorithm, we must apply heuristic rules to specify the order of servicing the victims. For that, at each discrete time point during the implementation of the algorithm, we compute a *victim utility score* (VUS) for each victim in the cluster of each ambulance. Formally, the utility score of a victim $v \in L_\lambda$ for an ambulance a_λ which is positioned at node l is the ratio of the weight of victim v (W_R or W_G) divided by the summation of travel time between nodes l and v (t_{lv}) and the treatment time of victim v (S_v). That is,

$$VUS_{lv}^{a_\lambda} = \begin{cases} \dfrac{W_R}{t_{lv} + S_v} & v \in R \\ \dfrac{W_G}{t_{lv} + S_v} & v \in G. \end{cases} \tag{13}$$

Using the above utility score, we iteratively dispatch each ambulance to a victim in its cluster with the highest VUS. In this way, we give higher priorities to red victims as well as the victims whose servicing them can be achieved in a shorter time (i.e., this rule reduces the weighted latest service completion times). We remark that the VUS for each victim and ambulance is updated dynamically over time depending on the location of that ambulance.

Once an ambulance a_λ reaches a victim, it checks whether the victim is red or green. If the victim is green, the ambulance a_λ treats the victim, removes the victim from its cluster L_λ, and moves to the victim on L_λ with the highest VUS. If the victim is red, the ambulance a_λ services the victim, then chooses a hospital with respect to a *hospital utility score* (HUS) as follows. To compute the utility of each hospital ($h \in H$) with remaining capacity, and for each ambulance a_λ that is positioned at the node of a red victim v_r, we incorporate (1) the travel time from the location of the victim to hospital h ($t_{v_r h}$), (2) the service time to drop off the red victim to that hospital (S_h), and (3) the remaining capacity of that hospital (C'_h) to compute the value of HUS as given in (14).

$$HUS^{a_\lambda}_{v_r h} = \frac{C'_h}{C_h \times (t_{v_r h} + S_h)}. \tag{14}$$

In the clustering algorithm, when a red victim is observed, once the values of HUS are identified, we send that ambulance with the red victim to the hospital with highest HUS for that ambulance. We note that in the design of the HUS, we consider both time (travel and service) as well as the remaining capacity of the available hospitals. In this way, we provide a fast service to the red victims while trying to keep the hospitals with low residual capacities available for servicing potential red victims that might be visited later and are potentially closer to those hospitals. We need to remark that although the HUS is inversely correlated with C_h on the one hand, it is directly correlated with C'_h on the other hand, i.e., the HUS is correlated with $\frac{C'_h}{C_h}$. Given that $C'_h = C_h$ for all $h \in H$ initially, the hospital capacities and the size of the hospitals have no effect on the HUS at the beginning of the operations. Once the transport of a red victim to a hospital is simulated, the available capacity of the destination hospital will be updated. After this, the delivered red victim will be removed from the cluster of its ambulance and then the ambulance moves to the node of the victim on L_λ with the highest VUS. This procedure is repeated for each ambulance until all victims in the cluster of that ambulance are serviced.

Similar to the optimisation-based algorithm, we let the ambulances which have completed servicing all the victims in their clusters, to service the remaining victims (i.e., the victims that are not serviced yet by the other ambulances) in a greedy manner with respect to VUS. Thus, we avoid having idle ambulances before all the victims are served and hence we improve the quality of our solution. A high-level pseudo-code of the clustering algorithm pointing out to only its major elements is presented in Algorithm 2.

Algorithm 2 Clustering algorithm

1: **Input:** an instance of the problem ▷ see Table 1
2: **Step 1:**
a: set $R = R^1$
b: set $G = V \setminus R^1$
c: set $S_v = 0$ for all $v \in V \setminus R^1 \cup G^1$
d: find cluster L_λ for all $a_\lambda \in A$ ▷ see Appendix and Section 5.2
3: **Step 2:**
a: set $V' = V$ ▷ define V' to keep track of victims who are not serviced yet
b: **dispatch** ambulance a_λ to service victims in L_λ with respect to VUS ▷ see equation (13)
c: **if** ambulance a_λ encounters a green victim v_g, **then:** ▷ case 1
d: service the green victim v_g
e: set $V' = V' \setminus \{v_g\}$
f: **end if**
g: **if** ambulance a_λ encounters a red victim v_r, **then:** ▷ case 2
h: choose h_* with respect to HUS ▷ see equation (14)
i: deliver v_r to h_*
j: update $C_{h_*} = C_{h_*} - 1$
k: set $H = H \setminus \{h_*\}$ if $C_{h_*} = 0$
l: set $V' = V' \setminus \{v_r\}$
m: **end if**
4: **Step 3:**
a: **if** ambulance a_λ completes servicing all victims in L_λ, **then:**
b: choose v_* for a_λ among the victims in V' with respect to VUS ▷ see equation (13)
c: dispatch a_λ to v_* and service it based on one of the two cases in step 2
d: set $V' = V' \setminus \{v_*\}$
e: **end if**
f: **if** $V' = \emptyset$, **then:** ▷ if all victims are serviced
g: **terminate** ▷ termination point
h: **end if**

5.3. Utility-based algorithm

In the utility-based algorithm, we assign the victims to the ambulances dynamically as new information reveals. With the motivation of prioritising the treatment of potential red victims (i.e., seriously injured victims who are not found yet by the ambulances), we include a feature in the utility-based algorithm which allows the ambulances to bypass green victims without servicing them in certain circumstances. This feature which can result in faster service completion times for more severely injured victims, can be a significant addition to the online algorithm in the context of mass casualty incidents where the locations of several victims are close to each other. We first describe this feature in detail and then explain the procedure of the utility-based algorithm.

5.3.1. Bypassing a green victim without providing service

To equip the utility-based algorithm with the feature mentioned above, once an ambulance reaches a green victim, we compute a *service utility score* (SUS) to decide whether to treat the green victim immediately or delay the treatment of the green victim to later visits. In here, we point out that a green victim that is not treated by the paramedics of an ambulance, will be added to a list called the *observed and untreated green victims* (OUG) list and will be serviced by the same or other ambulances later, e.g., when all potential red victims are treated. For an ambulance a_λ that is positioned at the node of a newly found green victim $v_g \in G$, the SUS is computed (online) with respect to (1) the proportion of known red victims to all victims with known triage (at the moment of decision making) denoted by Π_R, where $\Pi_R = \frac{|R^1|}{|R^1 \cup G^1|}$ at time 0, (2) the travel time ($t_{v_g v_*}$) between the current location of a_λ (v_g) and a node v_* with the highest $VUS^{a_\lambda}_{v_g v_*}$ (see Eq. (13)), (3) the revealed treatment time of the green victim S_{v_g}, (4) the weight of green victims W_G, and (5) the weight of red victims W_R such that:

$$SUS^{a_\lambda}_{v_g} = \frac{1}{1+\Pi_R} \times \frac{t_{v_g v_*}}{S_{v_g}} \times \frac{W_G}{W_R}. \tag{15}$$

In the utility-based algorithm, if $SUS^{a_\lambda}_{v_g} < 1$, the ambulance bypasses the green victim without providing treatment. Otherwise, the ambulance services the green victim immediately. In Eq. (15), the term $\frac{1}{1+\Pi_R}$ incorporates the real-time proportion of known red victims as a decision criteria for the algorithm. If Π_R is high, then SUS decreases and hence the likelihood of bypassing the green victim increases, i.e., if the ambulances would observe in the course of service operations that there are many red victims, the algorithm should likely bypass a green victim whereas if hardly any red victims have been observed in the process, the algorithm does not allow bypassing the green victims. Also, the term $\frac{t_{v_g v_*}}{S_{v_g}}$ in Eq. (15) ensures that green victims with relatively small treatment times have higher SUS, i.e., if the treatment time of the green victim (v_g) is relatively higher than the required travel time

to visit another victim (v_*) with unknown triage, then the SUS would be smaller for v_g. Furthermore, the term $\frac{W_G}{W_R}$ in Eq. (15) incorporates the relative weights of red and green victims in the decision making procedure such that the SUS for green victims would decrease where $\frac{W_G}{W_R}$ decreases, i.e., the likelihood of servicing green victims at their first visits decreases when the weight of red victims is significantly larger than the weight of green victims. To further elaborate on the bypassing feature of the utility-based algorithm, and to distinguish the scheduling procedure of this algorithm with the clustering and the optimisation-based algorithms, we apply a simple numerical example as follows. In this example, we consider a scenario during the implementation of the algorithms with a single ambulance and only two remaining victims v_1 and v_2 whose triages are not given to the ambulance a priori. We consider the scenario where the single ambulance a_1 is just arrived at v_1 and has realised that the victim is of green triage (i.e., $v_1 \in G$) with a treatment time of $S_{v_1} = 200$. In this scenario, we assume that no red victim is previously found such that $\Pi_R = 0$. Furthermore, we assume $W_G = 1$, $W_R = 10$, and the travel time between v_1 and v_2 is 400. Since the clustering and optimisation-based algorithms do not incorporate the bypassing feature, in these algorithms, the ambulance first services v_1 and then moves towards v_2. However, in the utility-based algorithm the service utility score is computed as $SUS_{v_1}^{a_1} = \frac{1}{1+\Pi_R} \times \frac{t_{v_1v_2}}{S_{v_1}} \times \frac{W_G}{W_R} = \frac{1}{1+0} \times \frac{400}{200} \times \frac{1}{10} = 0.2$. In this case, since $SUS_{v_1}^{a_1} = 0.2 < 1$, the ambulance bypasses v_1 immediately to service v_2, and adds v_1 to the set of observed but unserviced victims (OUG). In this way, the algorithm prioritises servicing red victims where there is a merit for such a prioritisation depending on the real time information, which leads to a reduced objective function value for this algorithm. We present the detailed procedure of the utility-based algorithm in the next subsection.

5.3.2. The procedure for the utility-based algorithm

We remark that the utility-based algorithm utilises various utility scores, namely, VUS and HUS defined in Section 5.2, and SUS presented in Section 5.3.1. The algorithm uses a list of unassigned victims denoted by V' to keep track of victims who are not assigned to any ambulances and initialises it by setting it equal to the list of all victims at the beginning. In addition, the algorithm defines and initialises an empty list to keep track of green victims which are observed but not treated due to their high treatment times, i.e., the OUG list. At the beginning, each ambulance is assigned to a victim with the highest VUS (see Eq. (13)). Then, the ambulances are dispatched to their assigned victims. Once an ambulance a_λ reaches to its assigned victim, the paramedics examine and learn the triage and treatment time of the victim and then share this information with all the other ambulances. In this sense, there is an important difference between the utility-based algorithm and the other two online algorithms. In those algorithms, since the victims were allocated to the ambulances before their conditions were examined, there was no need for information sharing between the paramedics on the ambulances as the same ambulance always had to serve the victims allocated to it. However, in the utility-based algorithm, once the triage and treatment time of a victim is examined, this information will be shared amongst the paramedics as some green victims will be bypassed initially and will be seen later. In the utility-based algorithm, when an ambulance arrives to a victim location, two cases may arise:

- The victim is green: in this case, we compute $SUS_{v_g}^{a_\lambda}$ (see Eq. (15)), the paramedics treat the victim if $SUS_{v_g}^{a_\lambda} > 1$. Next, the victim is removed from the list of unassigned victims and the ambulance is dispatched to a victim with the highest VUS for a_λ (see Eq. (13)). Otherwise if $SUS_{v_g}^{a_\lambda} < 1$, the green victim is added to the set of observed but untreated victims (OUG), and the ambulance is dispatched to a victim with the highest VUS.
- The victim is red: in this case, the paramedics provide the required treatment to the victim and the ambulance is dispatched to a hospital with the highest hospital utility score HUS (see Eq. (14)). Then, this information is shared with the rest of the ambulances and the capacity of the destination hospital is decreased by one.

Once an ambulance delivers a red victim to a hospital, it will be dispatched to a victim with the highest VUS for that ambulance. A high-level pseudo-code of the utility-based algorithm summarising its main features is presented in Algorithm 3.

Algorithm 3 Utility-based algorithm

Input: an instance of the problem ▷ see Table 1
Initiation:
  a: $V' = V$ ▷ set of unassigned victims
  b: $OUG = \emptyset$ ▷ OUG : set of Observed and Untreated Green victims
Step 1: while $V' \setminus OUG \neq \emptyset$**:**
  ▷ give priority to potential unserviced red victims by ignoring green victims in OUG
  a: **dispatch** each ambulance $a_\lambda \in A$ to a victim in $V' \setminus OUG$ with respect to VUS ▷ see equation (13)
  b: **if** an ambulance $a_\lambda \in A$ encounters a green victim v_g, **then:**
    c: update Π_R
    d: learn S_{v_g}
    e: compute $SUS_{v_g}^{a_\lambda}$ ▷ see equation (15)
    f: **if** $SUS_{v_g}^{a_\lambda} \geq 1$, **then:**
      g: service v_g ▷ service the green victim
      h: set $V' = V' \setminus \{v_g\}$
      i: set $OUG = OUG \setminus \{v_g\}$ ▷ if the green victim belongs to OUG, remove it from OUG
    j: **else:** ▷ if $SUS_{v_g}^{a_\lambda} < 1$
      k: set $OUG = OUG \cup \{v_g\}$ ▷ share information and add the green victim to OUG
    l: **end if**
  m: **end if**
  n: **if** an ambulance $a_\lambda \in A$ encounters a red victim v_r, **then:**
    o: update Π_R
    p: learn S_{v_r}
    q: service v_r
    r: set $V' = V' \setminus \{v_r\}$ ▷ service the red victim
    s: determine $h_* \in H$ with respect to HUS ▷ see equation (14)
    t: dispatch a_λ to deliver v_r to h_*
    u: set $C_{h_*} = C_{h_*} - 1$
    v: set $H = H \setminus \{h_*\}$ if $C_{h_*} = 0$
  w: **end if**
Step 2: while $V' \setminus OUG = \emptyset$ **and** $OUG \neq \emptyset$**:**
  a: **dispatch** each ambulance $a_\lambda \in A$ to a victim in OUG with respect to VUS ▷ see equation (13)
  b: **if** an ambulance $a_\lambda \in A$ arrives at the node of a green victim $v_g \in OUG$, **then:**
    c: service v_g
    d: set $V' = V' \setminus \{v_g\}$
    e: set $OUG = OUG \setminus \{v_g\}$
  f: **end if**
  g: **if** $V \setminus OUG = \emptyset$ **and** $OUG = \emptyset$, **then:**
    h: **terminate** ▷ termination point
  i: **end if**

6. Computational analysis of the algorithms

In this section we investigate the performance of our algorithms developed in Section 5. The experiments were coded and performed in Python 3.9 on a device with Intel Core i5 processor, 8 GB of RAM and

64-bit Windows 10 operating system. The mathematical models were solved using Gurobi 9 under an academic licence. In the following, we first give information about the data sets used in our computational experiments and then present the computational results.

6.1. Description of the data sets

In order to computationally analyse our algorithms, we tested them on the same instances from Talarico et al. (2015). We recall that their study addressed the offline version of our problem in which the triage levels of the victims as well as their treatment times are known in advance. Different from this offline problem, in our study, we have the triage and treatment time of only a subset of the victims (partial information). In the case where *no information* is considered, we do not have this information for any of the victims. The definitions of partial information and no information are also presented in Section 3.2. In our study, the triage and treatment time of unknown victims are only revealed after an ambulance arrives at the casualty location and the paramedics assess the condition of the victim. In Talarico et al. (2015), 324 benchmark instances were introduced with number of hospitals varying from 1 to 4 and number of ambulances varying from 1 to 35. In 108 instances, the number of victims was set to 10, in another 108 instances, the number of victims was set to 25 and in the last 108 instances, the number of victims was set to 50. Each of these instances were then solved four times by changing the ratio of the weight of the green victims to the weight of the red victims, i.e., $\frac{W_G}{W_R}$. In particular, the cases with $\frac{W_G}{W_R} = 1$, $\frac{W_G}{W_R} = \frac{1}{2}$, $\frac{W_G}{W_R} = \frac{1}{5}$ and $\frac{W_G}{W_R} = \frac{1}{10}$ were solved. Since W_G was set equal to 1 in these instances, they are equivalent to cases where W_R takes a value from 1, 2, 5 or 10. In our computational experiments, first we analyse the case with no information over all the 324 instances with varying values of W_R. We then investigate the cases where 20%, 40% and 60% of the victim conditions are known in advance (partial information).

6.2. Analysis of the performance of the algorithms

As it is stated in the previous sections, to test the performance of an online algorithm, the experimental competitive ratio notion is used. For an online algorithm on a particular instance, the obtained objective function value from that algorithm over the optimal objective function value of the offline problem gives the experimental competitive ratio of that algorithm for that specific instance. In here, we first present the results of our online algorithms on the cases where no information about the condition of the victims is known, and then present the results for the cases where the triage and required treatment times for some of the victims are given.

6.2.1. Results of the cases with no information

In this section, we compare the results obtained from our online algorithms with the offline objective function values which are extracted from Talarico et al. (2015) to report the experimental competitive ratios of our algorithms on the benchmark instances. We note that the focus of our study is not on solving the offline version of the problem. The detailed results for each of the cases with 10, 25 and 50 victims are given in Figs. 3–5, respectively.

In Fig. 3, the results of all the three algorithms are given over the first 108 instances with 10 victims, no information about the victim conditions, and for all the cases of W_R when it is equal to 1, 2, 5 or 10. Since in calculation of the experimental competitive ratio (denoted by ECR in the graphs), the ratio of the solution from the online algorithm over the optimal objective function value of the offline algorithm is considered, the value of the experimental competitive ratio for an online algorithm cannot be lower than 1.

As can be observed in Fig. 3, generally, the performance of the optimisation-based algorithm is slightly better compared to the other two algorithms, particularly for the first 40 instances. This is such that the average ECRs over the first 40 instances of the optimisation-based algorithm are 1.183 and 1.219 for $W_R = 1$ and $W_R = 2$, respectively. On the other hand, for $W_R = 1$ and $W_R = 2$, the average ECRs over the first 40 instances are 1.389 and 1.384 for the clustering and 1.348 and 1.406 for the utility-based algorithms. This trend does not follow for rest of the instances and the difference between the performance of the algorithms is not significant. Nevertheless, the average performance of the optimisation-based algorithm remains better than the other two algorithms when all the instances are considered as well. For example, the average ECRs of the algorithms over all the instances with $W_R = 1$ are 1.430, 1.389 and 1.332 for the clustering, utility and optimisation-based algorithms, respectively. The average ECRs over all the instances with 10 nodes and $W_R = 10$ are 1.449, 1.415 and 1.380 for the clustering, utility and optimisation-based algorithm which shows the initial observation that the optimisation-based algorithm is superior compared to the other two algorithms with 10 victims and no initial information.

Another observation can focus on the number of instances where each algorithm had the best performance among the three. In this regard, again, the optimisation-based algorithm was superior with 59, 52, 54 and 53 cases out of 108 where this algorithm found better solutions compared to the other two for varying W_R values from 1 to 10. While the performance of the optimisation-based is better on average, its drawback is its reliance on access to an optimisation solver. Moreover, when looking at the worst case performance (maximum ECR) of these algorithms over the 108 instances with 10 victims and no information on victim conditions, except for the case with $W_R = 2$, the utility-based algorithm has shown a better performance compared to both the optimisation and clustering algorithms.

Fig. 4 gives the same information as in Fig. 3 but for instances with 25 victims. Different from the instances with 10 victims, in the cases with 25 victims, when the number of victims increases, the optimisation-based algorithm does not seem to have a better performance compared to the other two algorithms. In fact, in these instances, the utility-based algorithm is superior in terms of both the average of the ECR values and the worst case performance identified by the maximum ECR. For example, the highest ECR value of the utility-based algorithm is 1.754, 2.171, 2.210 and 2.365 for $W_R = 1, 2, 5$ and 10, respectively. This shows that the utility-based algorithm shows a reliable performance even for the extreme cases when W_R is set to 10. On the other hand, the performance of the optimisation-based and clustering algorithms are not reliable when looking at their maximum ECR values. The maximum ECR of the optimisation-based and clustering algorithms are both within the instances with $W_R = 1$ where the maximum ECR is 4.620 for the optimisation-based and 3.545 for the clustering algorithm.

From an average performance perspective, the utility-based algorithm is superior within all the tested W_R values. That is for $W_R = 1$, the average ECR of the utility-based algorithm is 1.754 in comparison to 1.912 and 2.109 for clustering and optimisation-based algorithms. When $W_R = 2$, a similar comparison is observed and these values are 1.610, 1.761 and 1.912 for the utility, clustering and optimisation-based algorithms, respectively. For cases with $W_R = 5$ and 10, we also observe a similar pattern where the utility-based algorithm outperforms both clustering and optimisation-based algorithms. Recalling that the utility-based algorithm is the only algorithm that shows reaction to the observed information about the condition of the victims (e.g., bypassing green victims under certain situations), we point out that this consideration of new information results in its superior performance. A comparison between clustering and optimisation-based algorithms shows that, on average, the clustering algorithm has a better performance compared to the optimisation-based algorithm.

From the best case performance translated into finding the best solution over each instance, again the utility-based algorithm outperforms

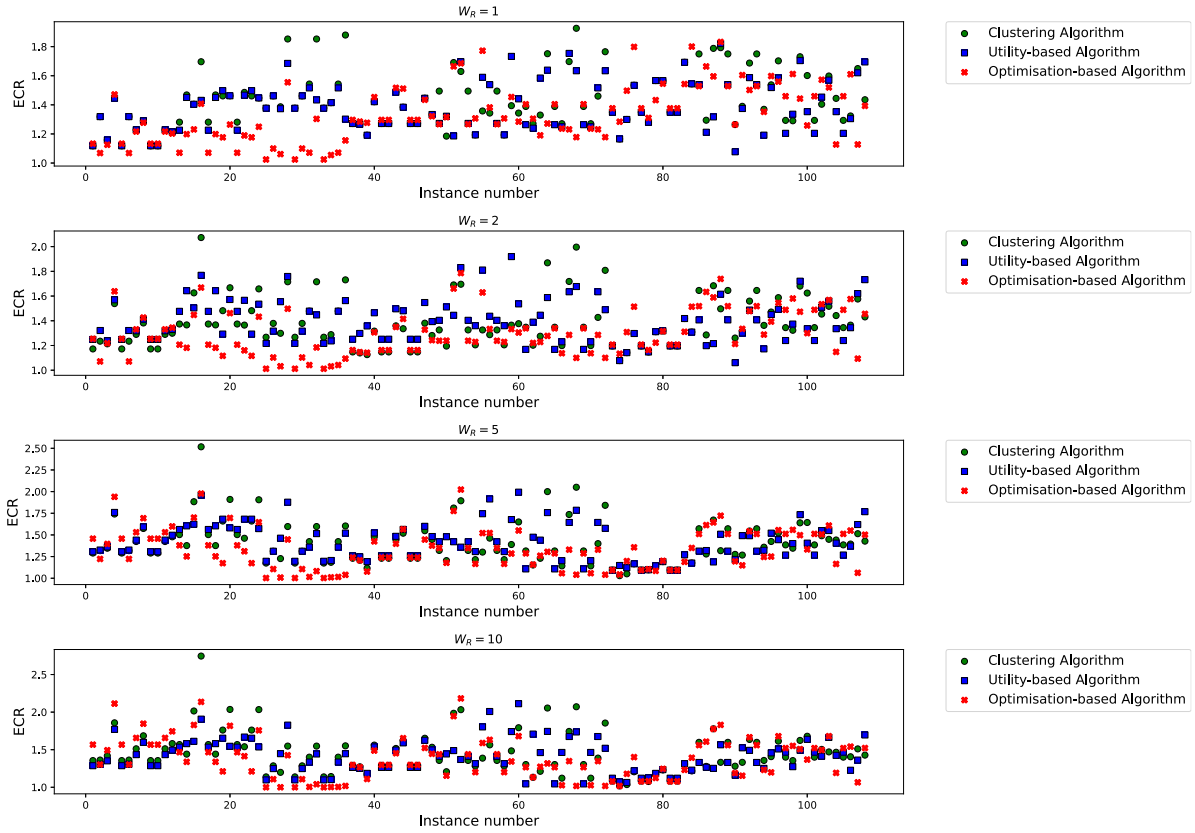


Fig. 3. Results for instances with 10 victims.

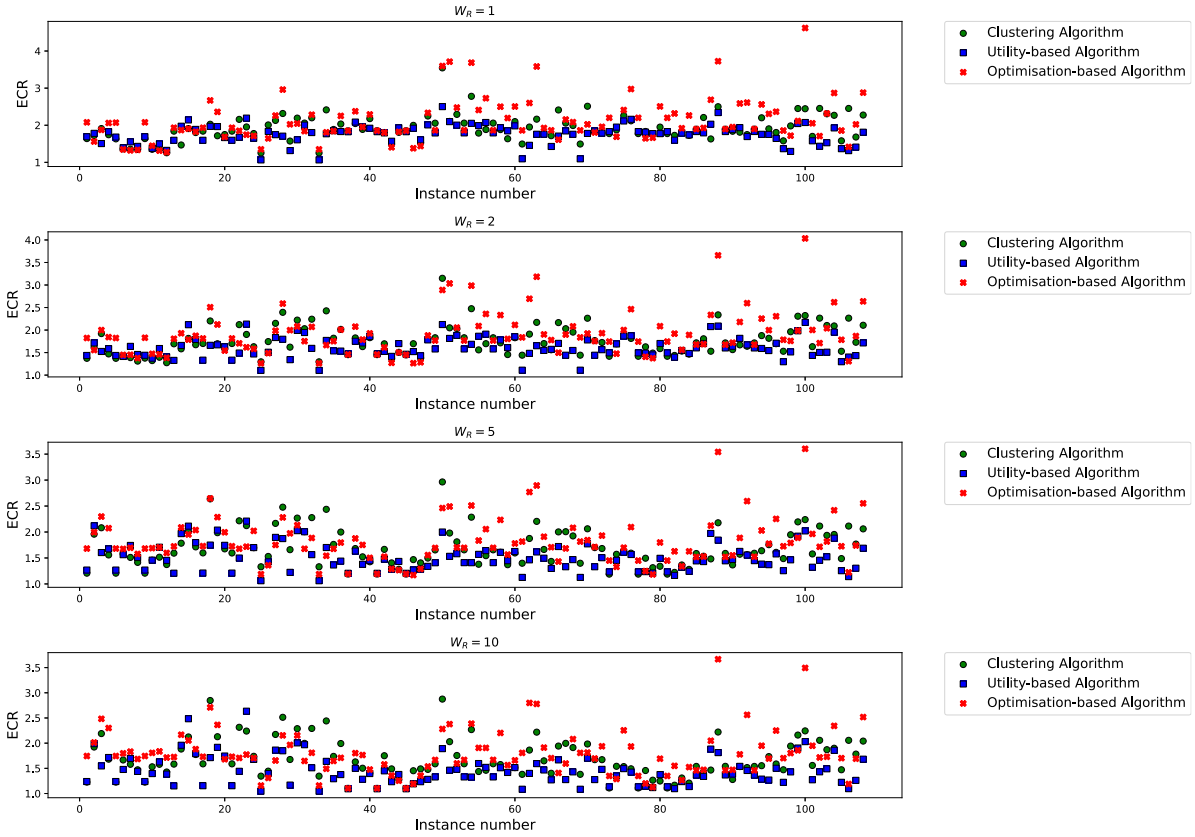


Fig. 4. Results for instances with 25 victims.

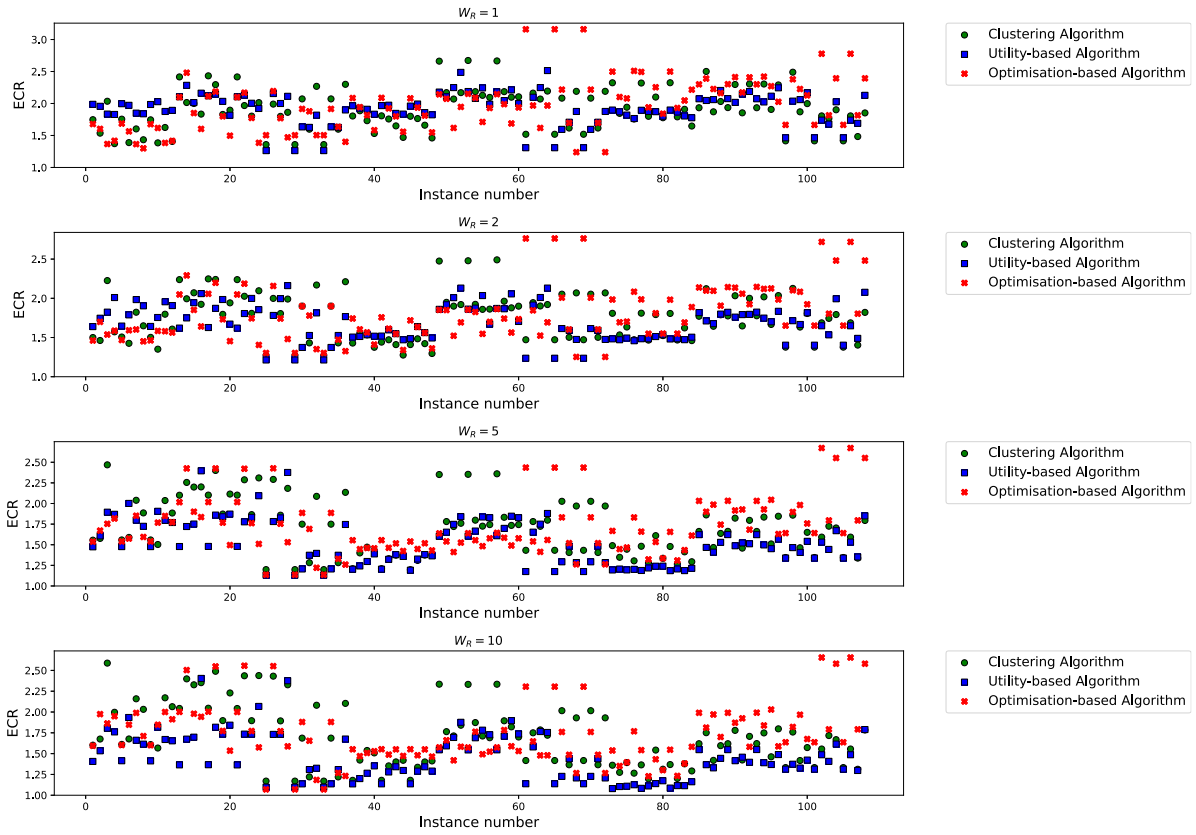


Fig. 5. Results for instances with 50 victims.

the other two algorithms. For the cases with $W_R = 1, 2, 5$ and 10, the utility-based algorithm finds the best solutions in 66, 57, 66 and 77 instances out of the 108 tested cases. For the remaining instances where the clustering or optimisation-based algorithms found better solutions, clustering algorithm found better solutions in more cases over all the values of W_R.

Fig. 5 gives the results of the instances with 50 victims. These are the largest instances that were tested in Talarico et al. (2015). In these larger instances, again the utility-based algorithm shows consistently better performance. Except for the case with $W_R = 1$ where the clustering algorithm with an average ECR of 1.905 in comparison to an average ECR of 1.919 achieved from the utility-based algorithm had a better performance, in rest of the cases, the average performance of the utility-based algorithm was superior compared to the other two algorithms. The superiority of the utility-based algorithm is more evident when the value of W_R increases. For example, when $W_R = 10$, the average ECR of the utility-based algorithm is 1.454 which is lower than an average of 1.693 for the clustering algorithm, and 1.723 for the optimisation-based algorithm. A comparison between the clustering and optimisation-based algorithms shows that the clustering algorithm outperforms the optimisation-based algorithm.

When looking at the worst performance amongst the instances, again, the utility-based algorithm outperforms the other two algorithms. For example, with $W_R = 5$, the highest ECR obtained from the utility-based algorithm is 2.396 but the highest ECR of the clustering and optimisation-based algorithms are 2.469 and 2.673, respectively. When looking at the number of times each of these algorithms found the best solution in the 108 tested instances, while with lower W_R values, the performance of the clustering and utility-based algorithms are almost similar, when the value of W_R increases, the performance of the utility-base algorithm is significantly better than the other two algorithms. In instances when $W_R = 5$, in 69 instances, and when $W_R = 10$, in 83 instances out of 108, the utility-based algorithm found better solutions compared to the other two algorithms. This again highlights the impacts of considering revealed information in the design of this online algorithm and shows the superiority of the utility-based algorithm which allows bypassing some of the green victims.

6.2.2. *Results of the cases with partial information*

In order to investigate the impact of having access to partial information about the conditions of the victims, we selected a total of 60 instances from each of the categories with 10, 25 and 50 victims (20 instances from each category) and tested our algorithms under three more scenarios. In these scenarios, we investigated cases where the triage levels and the treatment times of 20%, 40% and 60% of the victims are known in advance. Since for generating these scenarios, it matters that the partial information is available for which subset of the victims, for each of the instances, we generated 10 different samples. For example, for a specific instance with 25 victims where the conditions of 40% of them is available, we generated 10 samples in which the information of a different subset of victims is available in each of them. We note that the situation where no information is available (Section 6.2.1) can be categorised as 0% of information and does not require multiple samples. In the following, we analyse the performance of the three algorithms on all the selected instances with 10, 25 and 50 victims based on varying W_R values, i.e., $W_R = 1, 2, 5$ and 10.

Fig. 6 gives the summary of the results over all the selected instances when $W_R = 1$. As can be observed, in the case when $W_R = 1$, with more information, the optimisation-based algorithm tends to show a better performance. For example, in the case with 25 victims, when 60% of the information is available, the ECR is dropped to 1.63 from 2.05 (which is the ECR for the case when no information is available). An interesting observation is that when only 20% of the information is available, in none of the cases an improvement of the results is observed. This is perhaps because access to limited information cannot

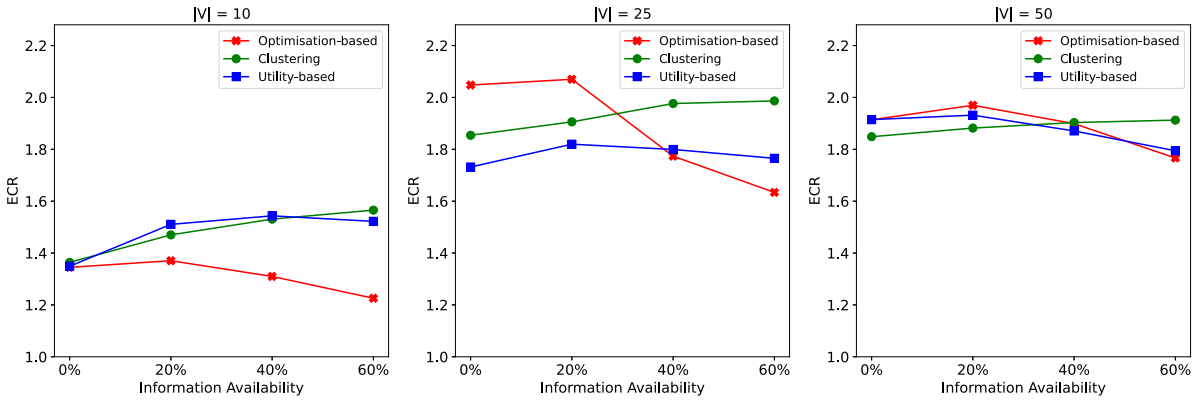


Fig. 6. Average results when $W_R = 1$.

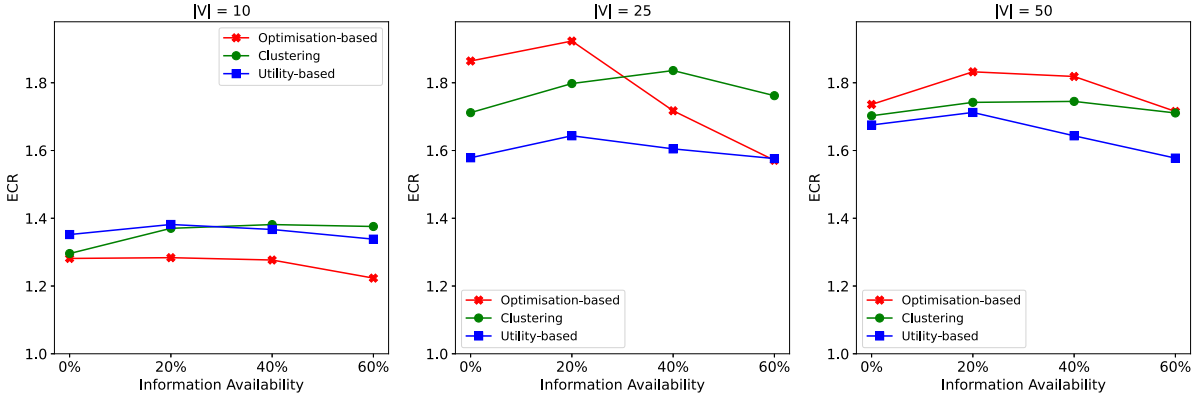


Fig. 7. Average results when $W_R = 2$.

help the algorithms in achieving better results. Another important observation is that when $W_R = 1$, access to more information does never guarantee a better solution for the clustering algorithm. The utility-based algorithm however, tends to find better solutions when more information is available. This is particularly correct for larger instances with 50 or 25 nodes.

Fig. 7 gives the results of the tested instances for the introduced algorithms when $W_R = 2$. In general, our algorithms show very good performances on average. For example, when $W_R = 2$, the worst ECR is from the optimisation-based algorithm over instances with 25 victims and 20% of information which is at only 1.92. In Fig. 7, we can see that the utility-based algorithm outperforms the clustering and optimisation-based algorithms for instances with 25 and 50 victims over all categories of information availability. In all the cases of the utility-based algorithm with different number of victims, access to 20% of information only worsens the found solutions. However, when more information is available in cases with 40% and 60% of information availability, the performance of the utility-based algorithm improves.

Fig. 8 gives the results of testing our algorithms on the same instances when $W_R = 5$. As can be observed, with these instances, all of our algorithms were again able to find good solutions and the worst case performance is even better than the case with $W_R = 2$ with the lowest ECR being at 1.82. While for the small instances with 10 victims, the performance of the optimisation-based algorithm is superior marginally, in the cases with 25 and 50 victims, the utility-based algorithm outperforms the other two algorithms significantly. When $W_R = 5$, for instances with 50 victims, even 20% of information helps the utility-based algorithm to obtain better solutions. The clustering algorithm also outperforms the optimisation-based algorithm over all the cases of information availability when $|V| = 50$.

Fig. 9, presents the results when $W_R = 10$. As it is evident in this figure, the utility-based algorithm outperforms the rest of the algorithms in all the cases. In all the cases with 10 and 50 victims, the utility-based algorithm continues to show a better performance with availability of more data. Since in a real-life scenario, the assigned weight to red victims (W_R), should be higher than green victims (W_G), we can see that the utility-based continuously outperforms the other two algorithms specially for larger instances. Another advantage of this algorithm is its scalability as it can solve the larger instances with 50 victims in merely 1 s.

As the weight of red victims (W_R) increases, the utility-based algorithm outperforms both optimisation-based and clustering algorithms. The reason behind this is that the utility-based algorithm incorporates a bypassing feature, allowing ambulances to navigate the network and locate red victims more efficiently. Consequently, these victims can be transported to hospitals more promptly compared to the optimisation-based and clustering algorithms. From a theoretical perspective, the bypassing feature of the utility-based algorithm reduces the latest service completion time of the red victims. Consequently, when the weight of red victims (i.e., W_R) increases, the objective function value of the utility-based algorithm would be less than the objective function value of the other two algorithms, i.e., when W_R increases, the utility-based algorithm gradually outperforms the optimisation-based and clustering algorithms.

Table 3 gives the summary of the performance of the optimisation-based, utility-based and clustering algorithms and their obtained ECR values from our computational analysis in this section. This table includes the mean and variance of the ECR values over different instances with 10, 25 and 50 victims and with varying information availability including 0%, 20%, 40% and 60%. As can be observed, the utility-based algorithm outperforms the other two algorithms on average and also shows a lower variance which confirms its robustness and superiority compared to the optimisation-based and clustering algorithms.

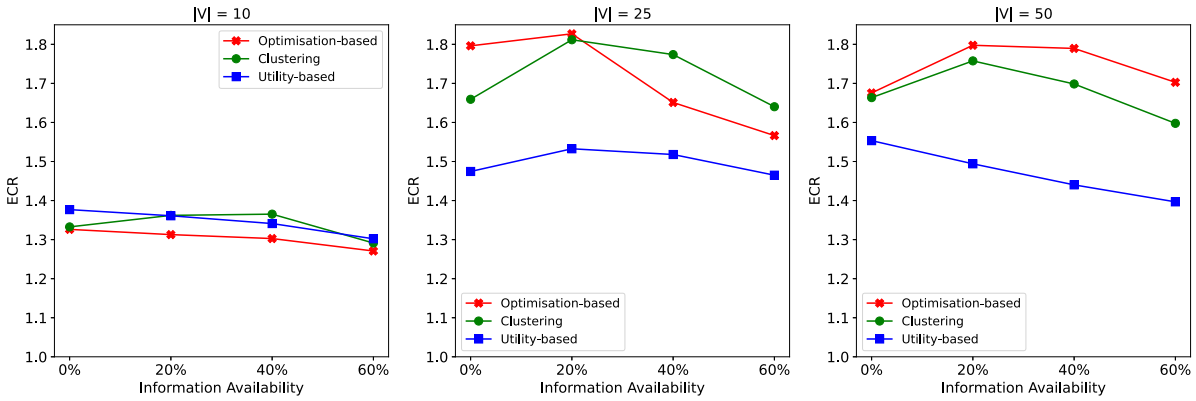


Fig. 8. Average results when $W_R = 5$.

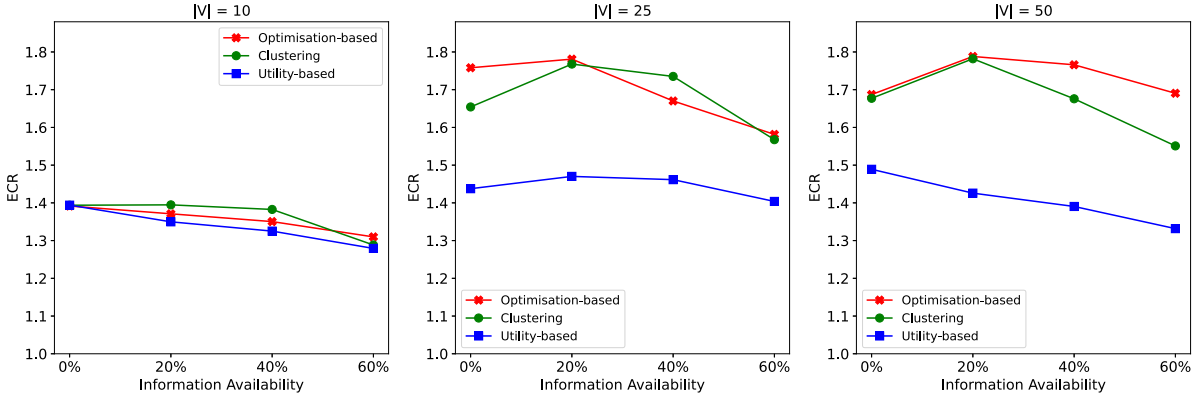


Fig. 9. Average results when $W_R = 10$.

Table 3
Summary of the results.

	Information availability	0%						20%						40%						60%					
	Algorithm	Optimisation-based		Clustering		Utility-based		Optimisation-based		Clustering		Utility-based		Optimisation-based		Clustering		Utility-based		Optimisation-based		Clustering		Utility-based	
W_R	$\|V\|$	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
1	10	1.34	0.13	1.36	0.18	1.35	0.18	1.37	0.17	1.47	0.17	1.51	0.18	1.31	0.13	1.53	0.20	1.54	0.18	1.23	0.10	1.57	0.25	1.52	0.20
	25	2.05	0.36	1.85	0.26	1.73	0.22	2.07	0.26	1.91	0.22	1.82	0.19	1.77	0.16	1.98	0.22	1.80	0.18	1.63	0.20	1.99	0.24	1.77	0.19
	50	1.91	0.42	1.85	0.34	1.91	0.28	1.97	0.33	1.88	0.32	1.93	0.28	1.90	0.26	1.90	0.33	1.87	0.28	1.77	0.24	1.91	0.37	1.79	0.28
2	10	1.28	0.10	1.30	0.13	1.35	0.18	1.28	0.10	1.37	0.15	1.38	0.12	1.28	0.11	1.38	0.19	1.37	0.15	1.22	0.10	1.38	0.23	1.34	0.16
	25	1.86	0.32	1.71	0.27	1.58	0.17	1.92	0.22	1.80	0.18	1.64	0.18	1.72	0.20	1.84	0.21	1.60	0.15	1.57	0.23	1.76	0.22	1.58	0.15
	50	1.74	0.34	1.70	0.30	1.68	0.24	1.83	0.31	1.74	0.26	1.71	0.28	1.82	0.25	1.74	0.28	1.64	0.24	1.72	0.24	1.71	0.29	1.58	0.22
5	10	1.33	0.17	1.33	0.17	1.38	0.22	1.31	0.14	1.36	0.19	1.36	0.16	1.30	0.13	1.37	0.20	1.34	0.14	1.27	0.11	1.29	0.20	1.30	0.15
	25	1.80	0.36	1.66	0.36	1.47	0.25	1.83	0.27	1.81	0.27	1.53	0.26	1.65	0.21	1.77	0.33	1.52	0.26	1.57	0.23	1.64	0.25	1.46	0.22
	50	1.68	0.32	1.66	0.33	1.55	0.29	1.80	0.30	1.76	0.37	1.49	0.23	1.79	0.25	1.70	0.30	1.44	0.21	1.70	0.24	1.60	0.28	1.40	0.19
10	10	1.39	0.22	1.39	0.20	1.39	0.24	1.37	0.18	1.39	0.22	1.35	0.18	1.35	0.17	1.38	0.22	1.33	0.16	1.31	0.14	1.29	0.21	1.28	0.17
	25	1.76	0.38	1.65	0.42	1.44	0.34	1.78	0.26	1.77	0.31	1.47	0.29	1.67	0.27	1.74	0.36	1.46	0.29	1.58	0.26	1.57	0.28	1.40	0.23
	50	1.69	0.33	1.68	0.37	1.49	0.28	1.79	0.30	1.78	0.39	1.43	0.22	1.77	0.27	1.68	0.34	1.39	0.21	1.69	0.27	1.55	0.29	1.33	0.18
	Avg	1.65	0.29	1.60	0.28	1.53	0.24	1.69	0.24	1.67	0.25	1.55	0.21	1.61	0.20	1.67	0.27	1.53	0.20	1.52	0.20	1.60	0.26	1.48	0.19

6.3. Computational running time of the algorithms

The computational running time of the clustering and the utility-based algorithms is less than one second on the largest tested instances. This is due to the fact that these algorithms are based on simple heuristic rules and do not rely on the optimisation solvers. As for the first phase of the optimisation-based algorithm, which relies on solving the mathematical formulation, the computational running time is set to one hour, i.e., a threshold by which the optimal solution is found on smaller instances, and after which the incumbent solution does not substantially improve on larger instances in our computational experiments. As for the second phase of the optimisation-based algorithm, the computational running time is less than one second similar to those of the clustering and utility-based algorithms. The computational running time for obtaining the offline optimal results can be found in Talarico et al. (2015).

7. Conclusions and future research directions

In this work, we investigated an online optimisation problem (i.e., ARP-DSOT) to find the routes and schedules of the ambulances in the aftermath of disasters and mass casualty incidents. The online parameters are considered to model the uncertainty associated with triage levels and treatment times of the victims. We analysed this problem from a theoretical worst-case competitive ratio perspective comprehensively and provided several lower bounds on the competitive ratio of online algorithms under different scenarios of partial and no information. Our lower bounds show the worst-case compensation of solving the problem under incomplete information.

Furthermore, to address real-life instances of the problem, we proposed three novel online algorithms. One of our algorithms (i.e., the optimisation-based algorithm) is a hybrid heuristic procedure which makes good use of an exact offline formulation to determine the routes and schedules of ambulances. As a fast alternative to the optimisation-based algorithm which does not require access to an optimisation solver, we also propose another heuristic (i.e., the clustering algorithm) which mainly relies on dividing the victims into clusters and allocating the victims in each cluster to an ambulance. Our third algorithm (i.e., the utility-based algorithm) is based on various novel problem specific heuristic policies and requires having complete communication between the ambulances as well as constant information sharing between the hospitals and ambulances.

We verify the performance of our online algorithms by comparing their solutions with the offline optimal solutions (which are provided under complete information) on a wide range of 1296 instances from the literature. In particular, we focus on two main cases of partial and no information and conduct an extensive sensitivity analysis to understand the performance of our algorithms under different scenarios. We discuss in detail the advantages and drawbacks of these algorithms based on our computational experiments. Based on our observations, on special scenarios with a low number of victims and a high percentage of partial information, the optimisation-based algorithm outperforms the other two approaches and can be utilised in real-world mass casualty scenarios. However, when the size of the problem grows, the utility-based algorithm achieves consistently good results against the other two alternatives. Furthermore, as the utility-based algorithm can be implemented in very low running times (i.e., less than 1 s), it can be directly applied in response to real-world mass casualty incidents.

From a theoretical competitive analysis point of view, providing deterministic and randomised online solutions which can compete with our provided lower bounds remains as a challenging open research problem considering the complexity of the problem setting of the ARP-DSOT.

CRediT authorship contribution statement

Davood Shiri: Conceptualisation, Formal analysis, Methodology, Software, Writing. **Vahid Akbari:** Conceptualisation, Data Curation, Software, Visualisation, Writing. **Hakan Tozan:** Conceptualisation, Resources, Investigation.

Data availability

Data will be made available on request.

Appendix. The pseudo-code of the clustering procedure used in the clustering algorithm

In the following, we present the detailed pseudo-code that was used to develop the clustering algorithm presented in Section 5.2.

Algorithm 4 The clustering procedure for the clustering algorithm

Input: an instance of problem with partial information, i.e., sets R^1 and G^1 ▷ see Table 1
Initiation:
a: $L_\lambda = \emptyset$ ▷ set of assigned victims to ambulance $a_\lambda \in A$
b: $A' = A$ ▷ copy of the set of ambulances
c: $TL = R^1$ ▷ temporary list for assignment of victims to ambulances
if $R^1 = \emptyset$ **then:**
  $TL = V \setminus (R^1 \cup G^1)$
  go to 13
  if $V \setminus (R^1 \cup G^1) = \emptyset$ **then:**
    $TL = G^1$
    go to 13
  end if
else:
  go to 13
end if
if $A' = \emptyset$ **then:** ▷ if the assignment of victims to the ambulances is balanced
  set $A' = A$
  go to 19
  ▷ add all ambulances to A' in order to assign a new victim to each ambulance
else:
  go to 19
end if
determine $a_* \in A'$ ▷ a_* is the first indexed ambulance in A'
find $v_* \in TL$ ▷ v_* is the victim with the closest distance to a_*
add v_* to L_* ▷ add v_* to the list of assigned victims to ambulance a_*
set $A' = A' \setminus \{a_*\}$
set $V = V \setminus \{v_*\}$
if $v_* \in R^1$ **then:** set $R^1 = R^1 \setminus v_*$ **end if**
if $v_* \in V \setminus (R^1 \cup G^1)$ **then:** set $V \setminus (R^1 \cup G^1) = V \setminus (R^1 \cup G^1 \cup \{v_*\})$ **end if**
if $v_* \in G^1$ **then:** set $G^1 = G^1 \setminus v_*$ **end if**
if $V = \emptyset$ **then:** ▷ if all the victims are assigned to the ambulances
  terminate ▷ end of clustering
else:
  Go to 3
end if

References

Ajam, Meraj, Akbari, Vahid, Salman, F. Sibel, 2022. Routing multiple work teams to minimize latency in post-disaster road network restoration. European J. Oper. Res. 300 (1), 237–254.

Akbari, Vahid, Shiri, Davood, 2021. Weighted online minimum latency problem with edge uncertainty. European J. Oper. Res. 295 (1), 51–65.
Akbari, Vahid, Shiri, Davood, 2022. An online optimization approach for post-disaster relief distribution with online blocked edges. Comput. Oper. Res. 137, 105533.
Akbari, Vahid, Shiri, Davood, Sibel Salman, F., 2021. An online optimization approach to post-disaster road restoration. Transp. Res. B 150, 1–25.
Anuar, Wadi Khalid, Lee, Lai Soon, Pickl, Stefan, Seow, Hsin-Vonn, 2021. Vehicle routing optimisation in humanitarian operations: A survey on modelling and optimisation approaches. Appl. Sci. 11 (2).
Aringhieri, Roberto, Bigharaz, Sara, Duma, Davide, Guastalla, Alberto, 2022. Fairness in ambulance routing for post disaster management. CEJOR Cent. Eur. J. Oper. Res. 30, 189–211.
Aringhieri, R., Bruni, M.E., Khodaparasti, S., van Essen, J.T., 2017. Emergency medical services and beyond: Addressing new challenges through a wide literature review. Comput. Oper. Res. 78, 349–368.
Ashlagi, Itai, Burq, Maximilien, Dutta, Chinmoy, Jaillet, Patrick, Saberi, Amin, Sholley, Chris, 2022. Edge-weighted online windowed matching. Math. Oper. Res..
Azar, Yossi, Epstein, Leah, 1998. On-line machine covering. J. Sched. 1 (2), 67–77.
Ben-David, Shai, Borodin, Allan, Karp, Richard, Tardos, Gabor, Wigderson, Avi, 1994. On the power of randomization in on-line algorithms. Algorithmica 11 (1), 2–14.
Bonifaci, Vincenzo, Lipmann, Maarten, Stougie, Leen, et al., 2006. Online Multi-Server Dial-a-Ride Problems. TU/e, Eindhoven University of Technology, Department of Mathematics and ....
Borodin, Allan, El-Yaniv, Ran, 2005. Online Computation and Competitive Analysis. cambridge University Press.
Büttner, Sabine, Krumke, Sven O., 2016. The Canadian tour operator problem on paths: tight bounds and resource augmentation. J. Combin. Optim. 32, 842–854.
Caragiannis, Ioannis, Kaklamanis, Christos, Papaioannou, Evi, 2008. Competitive algorithms and lower bounds for online randomized call control in cellular networks. Netw.: Int. J. 52 (4), 235–251.
Chen, Peng Will, Nie, Yu Marco, 2015. Optimal transit routing with partial online information. Transp. Res. B 72, 40–58.
Chen, Cong, Penna, Paolo, Xu, Yinfeng, 2020. Online scheduling of jobs with favorite machines. Comput. Oper. Res. 116, 104868.
CRED & UNDRR, 2020. Human Cost of Disasters: An Overview of the Last Twenty Years 2000–2019. Tech. Rept., Centre for Research on the Epidemiology of Disasters and UN Office for Disaster Risk Reduction.
De la Torre, Luis E., Dolinskaya, Irina S., Smilowitz, Karen R., 2012. Disaster relief routing: Integrating research and practice. Soc.-Econ. Plan. Sci. 46 (1), 88–97, Special Issue: Disaster Planning and Logistics: Part 1.
Dwibedy, Debasis, Mohanty, Rakesh, 2022. Semi-online scheduling: A survey. Comput. Oper. Res. 139, 105646.
Epstein, Leah, 2009. On online bin packing with LIB constraints. Nav. Res. Logist. 56 (8), 780–786.
Farahani, Reza Zanjirani, Lotfi, M.M., Baghaian, Atefe, Ruiz, Rubén, Rezapour, Shabnam, 2020. Mass casualty management in disaster scene: A systematic review of OR & MS research in humanitarian operations. European J. Oper. Res. 287 (3), 787–819.
Ghiani, Gianpaolo, Guerriero, Francesca, Laporte, Gilbert, Musmanno, Roberto, 2003. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. European J. Oper. Res. 151 (1), 1–11.
He, Xiaozhou, Xiang, Jie, Xiao, Jin, Cheng, TCE, Tian, Yuhang, 2022. An online algorithm for the inventory retrieval problem with an uncertain selling duration, uncertain prices, and price-dependent demands. Comput. Oper. Res. 105991.
Hertrich, Christoph, Weiß, Christian, Ackermann, Heiner, Heydrich, Sandy, Krumke, Sven O, 2022. Online algorithms to schedule a proportionate flexible flow shop of batching machines. J. Sched. 1–15.
Hertz, Alain, Montagné, Romain, Gagnon, François, 2018. Online algorithms for the maximum k-colorable subgraph problem. Comput. Oper. Res. 91, 209–224.
Jaillet, Patrick, Lu, Xin, 2014. Online traveling salesman problems with rejection options. Networks 64 (2), 84–95.
Jaillet, Patrick, Stafford, Matthew, 2001. Online searching. Oper. Res. 49 (4), 501–515.
Jaillet, Patrick, Wagner, Michael R., 2006. Online routing problems: Value of advanced information as improved competitive ratios. Transp. Sci. 40 (2), 200–210.
Jaillet, Patrick, Wagner, Michael R., 2008. Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses. Oper. Res. 56, 745–757.
Kasaei, Maziar, Salman, F. Sibel, 2016. Arc routing problems to restore connectivity of a road network. Transp. Res. E 95, 177–206.
Lee, Yu-Ching, Chen, Yu-Shih, Chen, Albert Y., 2022. Lagrangian dual decomposition for the ambulance relocation and routing considering stochastic demand with the truncated Poisson. Transp. Res. B 157, 1–23.
Legrain, Antoine, Jaillet, Patrick, 2016. A stochastic algorithm for online bipartite resource allocation problems. Comput. Oper. Res. 75, 28–37.
Liu, Hui, Queyranne, Maurice, Simchi-Levi, David, 2005. On the asymptotic optimality of algorithms for the flow shop problem with release dates. Nav. Res. Logist. 52 (3), 232–242.
Luo, Ying, Schonfeld, Paul, 2011. Online rejected-reinsertion heuristics for dynamic multivehicle dial-a-ride problem. Transp. Res. Rec. 2218 (1), 59–67.
Moreno, Alfredo, Munari, Pedro, Alem, Douglas, 2019. A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration. European J. Oper. Res. 275 (1), 16–34.
Ojeda Rios, Brenner Humberto, Xavier, Eduardo C., Miyazawa, Flávio K., Amorim, Pedro, Curcio, Eduardo, Santos, Maria João, 2021. Recent dynamic vehicle routing problems: A survey. Comput. Ind. Eng. 160, 107604.
Oksuz, Mehmet Kursat, Satoglu, Sule Itir, 2020. A two-stage stochastic model for location planning of temporary medical centers for disaster response. Int. J. Disaster Risk Reduct. 44, 101426.
Pillac, Victor, Gendreau, Michel, Guéret, Christelle, Medaglia, Andrés L., 2013. A review of dynamic vehicle routing problems. European J. Oper. Res. 225 (1), 1–11.
Rabbani, Masoud, Oladzad-Abbasabady, Nastaran, Akbarian-Saravi, Niloofar, 2021. Ambulance routing in disaster response considering variable patient condition: NSGA-II and MOPSO algorithms. J. Ind. Manage. Optim..
Salman, F. Sibel, Gül, Sezer, 2014. Deployment of field hospitals in mass casualty incidents. Comput. Ind. Eng. 74, 37–51.
Sayarshad, Hamid R., Du, Xinpi, Gao, H. Oliver, 2020. Dynamic post-disaster debris clearance problem with re-positioning of clearance equipment items under partially observable information. Transp. Res. B 138, 352–372.
Schilde, M., Doerner, K.F., Hartl, R.F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res. 38 (12), 1719–1730.
Shiri, Davood, Akbari, Vahid, Salman, F. Sibel, 2020. Online routing and scheduling of search-and-rescue teams. OR Spectrum 42 (3), 755–784.
Shiri, Davood, Salman, F. Sibel, 2019. Competitive analysis of randomized online strategies for the online multi-agent k-Canadian traveler problem. J. Combin. Optim. 37, 848–865.
Shiri, Davood, Salman, F. Sibel, 2020. Online optimization of first-responder routes in disaster response logistics. IBM J. Res. Dev. 64, 1–9.
Shiri, Davood, Tozan, Hakan, 2022. Online routing and searching on graphs with blocked edges. J. Combin. Optim. 44 (2), 1039–1059.
Sleator, Daniel, Tarjan, Robert, 1985. Amortized efficiency of list update and paging rules. Commun. ACM 28, 202—208.
Soeffker, Ninja, Ulmer, Marlin W., Mattfeld, Dirk C., 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. European J. Oper. Res. 298 (3), 801–820.
Talarico, Luca, Meisel, Frank, Sörensen, Kenneth, 2015. Ambulance routing for disaster response with patient groups. Comput. Oper. Res. 56, 120–133.
Talebi, Ehsan, Shaabani, Mahnaz, Rabbani, Masoud, 2021. Bi-objective model for ambulance routing for disaster response by considering priority of patients. Int. J. Supply Oper. Manage..
Tikani, Hamid, Setak, Mostafa, 2019. Ambulance routing in disaster response scenario considering different types of ambulances and semi soft time windows. J. Ind. Syst. Eng. 12 (1), 95–128.
Tippong, Danuphon, Petrovic, Sanja, Akbari, Vahid, 2022. A review of applications of operational research in healthcare coordination in disaster management. European J. Oper. Res. 301 (1), 1–17.
Tlili, Takwa, Abidi, Sofiene, Krichen, Saoussen, 2018. A mathematical model for efficient emergency transportation in a disaster situation. Am. J. Emerg. Med. 36 (9), 1585–1590.
Tlili, Takwa, Harzi, Marwa, Krichen, Saoussen, 2017. Swarm-based approach for solving the ambulance routing problem. Procedia Comput. Sci. 112, 350–357, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.
Wang, Hao, Yan, Zhenzhen, Bei, Xiaohui, 2022. A non-asymptotic analysis for re-solving heuristic in online matching. Prod. Oper. Manage..
Xu, Jin, Gautam, Natarajan, 2020. On competitive analysis for polling systems. Nav. Res. Logist. 67 (6), 404–419.
Yao, Andrew Chi-Chih, 1977. Probabilistic computations: Towards a unified measure of complexity. In: Proceedings of the 18th Annual IEEE Symposium on the Foundations of Computer Science. pp. 222—227.
Yoon, Soovin, Albert, Laura A., 2020. A dynamic ambulance routing model with multiple response. Transp. Res. E 133, 101807.
Zhang, Huili, Luo, Kelin, Xu, Yao, Xu, Yinfeng, Tong, Weitian, 2022. Online crowdsourced truck delivery using historical information. European J. Oper. Res. 301 (2), 486–501.
Zhang, Huanan, Shi, Cong, Qin, Chao, Hua, Cheng, 2016. Stochastic regret minimization for revenue management problems with nonstationary demands. Nav. Res. Logist. 63 (6), 433–448.
Zhang, Huili, Tong, Weitian, Lin, Guohui, Xu, Yinfeng, 2019. Online minimum latency problem with edge uncertainty. European J. Oper. Res. 273, 418–429.
Zidi, Issam, Al-Omani, Mohammad, Aldhafeeri, Karim, 2019. A new approach based on the hybridization of simulated annealing algorithm and tabu search to solve the static ambulance routing problem. Procedia Comput. Sci. 159, 1216–1228, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019.