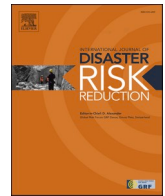




Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



## Multi-resource scheduling and routing for emergency recovery operations

Behrooz Bodaghi<sup>a,b</sup>, Shahrooz Shahparvari<sup>b,\*</sup>, Masih Fadaki<sup>b</sup>, Kwok Hung Lau<sup>b</sup>, Palaneeswaran Ekambaram<sup>a</sup>, Prem Chhetri<sup>b</sup>

<sup>a</sup> Faculty of Science, Engineering, and Technology, Swinburne University of Technology, Hawthorn, Australia

<sup>b</sup> School of Accounting, Information systems, and Supply chain, RMIT University, Melbourne, VIC, Australia

### ARTICLE INFO

#### Keywords:

Expendable resources  
Non-expendable resources  
Emergency recovery operations  
Multi-resource scheduling  
Heuristics algorithms  
Clustering algorithm  
Coronavirus outbreak case

### ABSTRACT

Efficient delivery of multiple resources for emergency recovery during disasters is a matter of life and death. Nevertheless, most studies in this field only handle situations involving single resource. This paper formulates the Multi-Resource Scheduling and Routing Problem (MRSRP) for emergency relief and develops a solution framework to effectively deliver expendable and non-expendable resources in Emergency Recovery Operations. Six methods, namely, Greedy, Augmented Greedy, k-Node Crossover, Scheduling. Monte Carlo, and Clustering, are developed and benchmarked against the exact method (for small instances) and the genetic algorithm (for large instances). Results reveal that all six heuristics are valid and generate near or actual optimal solutions for small instances. With respect to large instances, the developed methods can generate near-optimal solutions within an acceptable computational time frame. The Monte Carlo algorithm, however, emerges as the most effective method. Findings of comprehensive comparative analysis suggest that the proposed MRSRP model and the Monte Carlo method can serve as a useful tool for decision-makers to better deploy resources during emergency recovery operations.

### 1. Introduction

In emergency situations, such as major earthquakes or widespread outbreaks of infectious disease, quickly supplying diverse and critically needed rescue resources from various relief centres to many demand points can be a matter of life and death. A typical example is delivering urgent medical treatments/supplies from emergency recovery operation (ERO) control centres to various hospitals in a short period of time to treat a large number of injured patients when a destructive natural disaster strikes. For instance, in the ERO following the attack of Hurricane Katrina on New Orleans, USA, in 2005, delays in the treatment of victims occurred as a result of a shortage of nurses even though the medical supplies were distributed on time to those who needed them [1]. In another example, on Saturday, February 7, 2009, also known as Black Saturday, over 400 bushfires swept through parts of rural Victoria, Australia. The day is remembered for the deadliest and most devastating bushfires in the nation's history, with 173 deaths and 414 injuries. However, the Victorian emergency response teams' efforts were hampered by the fact that medical services were unavailable in local areas and some first aid services were uncoordinated in their initial response. It was reported that resources such as medical teams and

medical supplies were poorly coordinated, especially in providing relief to smaller, remote communities and people who stayed on their properties [2]. Such delay and poor coordination could be avoided or minimised if scheduling of both expendable and non-expendable resources could be taken into consideration in the same optimisation process. However, scheduling problems for multiple resources during emergency situations have become more complicated than in the case of single resource scheduling [1,3,4]. The ERO will be more complex considering the fact that relief centre (RC) is usually subject to severe resource limitations and acute time pressures in deciding what resources are to be sent to which hospitals to rescue/save lives.

Since time is critical in emergency situations and short response time can help save more lives, appropriate schedule and routing plans for emergency vehicles in delivering multiple resources from RCs to hospitals are of utmost importance. Reducing human casualties and fatalities is critically dependent on the rapid deployment of resources for EROs at demand points. Failure to assign adequate resources in a timely manner has been the root cause of many disastrous situations [3,5–7]. This is a challenging task given the scarceness of resources and the uncertain scale of the disasters [8,9]. Hospitals in a city are usually dispersed in geographical location and diverse in their ability to handle a

\* Corresponding author.

E-mail address: [shahrooz.shahparvari@rmit.edu.au](mailto:shahrooz.shahparvari@rmit.edu.au) (S. Shahparvari).

<https://doi.org/10.1016/j.ijdr.2020.101780>

Received 7 January 2020; Received in revised form 28 April 2020; Accepted 21 July 2020

Available online 29 August 2020

2212-4209/© 2020 Elsevier Ltd. All rights reserved.

surge capacity. An unexpected flood of patients can overwhelm hospitals and exceed their surge capacities, thereby affecting their preparedness for responding to emergencies and disasters [10,11].

ERO often mandates many types of resources to serve the demand points. Both expendable resources, such as food, clothing, shelter, and medical items, and non-expendable resources, such as nurses, surgeons, and medical equipment, are required in a devastation scenario in order to reduce the number of casualties. Among the resources that can be delivered from RCs, some are expendable, such as medical supplies, which can be simply unloaded at the destination so that the delivery vehicle can continue with its journey to other stops. The unloading of these resources will not impact much on the total completion time of the trip. However, some resources are non-expendable, for instance specialist surgeons and special medical equipment, which can be needed at multiple locations. As such, the delivery vehicle will have to wait until the relief operation at one stop is finished before it can deliver the resources to another stop to provide relief. The wait can significantly impact on the total journey time of the vehicle hence its opportunities to operate another run. This being the case, optimisation of routes and schedules of vehicles supplying multiple resources to different locations, referred to as the Multi-Resource Routing and Scheduling Problem (MRRSP) for emergency relief, is paramount for disaster management to be effective. To the best of the authors' knowledge, no previous studies have contextualised the ERO problem to address key operational challenges, such as timely distribution, by developing mechanisms to facilitate the coordination among resources. Both expendable and non-expendable items need to be examined with a consideration of collaboration between resources using various heuristics and clustering algorithms.

This paper, hence, aims to develop algorithms as a solution framework so that the emergency authorities can implement reasonably good to excellent solutions that generate, allocate and schedule resources for the EROs at an entire set of demand points in a disaster relief scenario within acceptable computational time. Hence, in this study, the MRRSP is modelled for disaster management centres to route and schedule emergency vehicles, in order to deliver expendable and non-expendable resources to hospitals. It will take into account the linear programming optimisation model, aiming to determine the routing of the demand points (e.g., hospitals) visited by chosen vehicle(s) to deliver requisite resources. At the same time, it will minimise completion times of recovery operations at individual demand points weighted by the severity level of EROs in each demand point.

Since MRRSP is related to the multiple Travelling Salesman Problem (mTSP) [12,13], involving both the routing and scheduling problems has proved to be NP-hard [14,15]. An efficient way to find a near-optimal schedule for multiple resources in emergency response facilitates the eventual attainment of effective resource management and ensures the welfare of disaster victims [16]. Six heuristics algorithms, namely, Greedy, Augmented Greedy, k-Node Crossover, Scheduling, Monte Carlo, and Clustering have been developed to obtain near-optimal solutions for the MRRSP. The heuristics are first validated, using small cases with only a few RCs, demand points and vehicles, against the exact solutions. Upon validation, ten large cases with a maximum of 15 RCs, 500 demand points and 50 vehicles, are implemented to evaluate the performance of the heuristics by comparing the solutions against those of the well-known genetic algorithm - a meta-heuristic approach.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature on vehicle routing and scheduling in EROs and identifies the research gap that this study attempts to address. Section 3 details the statement and formulation of the MRRSP. Section 4 shows the pseudo codes of the six heuristics used in this study. Section 5 empirically validates the six heuristics against the exact solutions for small instances. It then compares the performance of the six heuristics against that of the genetic algorithm for large instances and identifies the best heuristic. Section 6 concludes the study with a discussion on its

limitations and what future research directions could pursue.

## 2. Related works

This study falls in the field of logistics and transportation operations spanning across the categories of routing or scheduling optimisation and emergency relief. There has been a substantial amount of work on vehicle routing and scheduling for emergency relief or related problems using mathematical programming. Some of the latest works include Pillac et al. [17]; Shahparvari et al. [18]; Bruni et al. [19]; Zhang et al. [20] and Schneider and Nurre [21].

In routing and scheduling, for example, Bertazzi et al. [22] develop a mixed-integer programming model to solve the Multi-Depot Inventory Routing Problem which minimises routing costs by determining how to serve the customers from different depots while managing their inventory levels to avoid stock-outs. A three-phase meta-heuristic algorithm is designed to solve the problem. The proposed algorithm performs better than a branch-and-cut algorithm for this type of problem. Li et al. [23] apply Whittle's restless bandits approach to develop an index policy to optimize the scheduling of limited resources to a large number of jobs, such as medical treatments, with uncertain lifetimes and service times, in the aftermath of a mass casualty incident. The new approach enables the development of state-dependent policy to prioritise jobs to better utilise the resource.

Jung et al. [24] develop a model for allocating operating room capacity in hospitals to scheduled patients while accommodating randomly arrived emergency patients without incurring excessive delays. The aim is to develop a framework for aggregating weekly schedules and generating detailed daily schedules that minimise the total cost of the expected operating time, idle time, and overtime of the operating rooms. Using the proposed approach, the optimal capacity allocation of operating rooms to elective patients can be determined as a function of the emergency arrival rate. Mahmoudi et al. [25] develop a new network data envelopment analysis model for the sustainable Urban Network Design Problem. By integrating data envelopment analysis, game theory and sustainability requirements, a bi-level model is proposed for selecting and scheduling urban road construction projects which is solved using a meta-heuristic algorithm. The model improves the environmental and social performance of the urban network during all evaluation periods. Shi et al. [26] formulate a model for a Home Health Care Routing and Scheduling Problem from the perspective of Robust Optimisation (RO) to take into account uncertain travel and service times. The non-deterministic variables are defined based on the theory of budget uncertainty and arrival time of each caregiver is rewritten as a complicated recursive function. The model overcomes the shortcomings of many deterministic models for this problem.

In emergency relief research, Bababeik et al. [27] investigate the optimal location and allocation of relief trains to enhance the resilience level of the rail network. Using link exposure measure which considers the operational attributes of links and accessibility to road system, a bi-objective programming model is formulated and solved using an augmented e-constraint method combined with a fuzzy-logic approach. The proposed framework shows superiority in providing an economical and effective layout compared to conventional maximal covering model. Wu et al. [28] propose to use microblogging data to identify water-logging locations during urban rainstorm to optimize post-disaster rescue routes. By developing a location-routing problem model and solving it with NSGA-III algorithm, new routes to dissipate road traffic can be assigned timely which can be of great significance to the emergency rescue of victims during rainstorm disasters. The approach is novel in that it leverages social media data to obtain critical information on disasters to facilitate emergency relief. Zhu et al. [29] consider equity and priority issues in optimizing routing of rescue vehicles during emergency relief. Two models are developed taking into account the different levels of injury representing equity and the in-transit tolerable suffering duration representing rescue priority. The models excel in the

**Table 1**  
Summary of reviewed vehicle routing and scheduling optimisation models for emergency relief.

Study	Resource	Sequence	Mode	Objective	Solution	Model Limitation
Chang et al. [36]	Expendable	–	Stochastic	To minimise transportation cost, facility set up cost, equipment cost	Exact method	– Sequencing and scheduling of resources are not considered
Balcik et al. [37]	Expendable	Known	Stochastic	To maximise total expected demand	Exact method	– Assumes sequence of visit is known – The only sequencing/routing decision is whether the node should be visited each day
Huang et al. [45]	Expendable	Unknown	Deterministic	To minimise total travel time, sums of arrival times and unsatisfied demand	Heuristic method	– Considers only a single type of expendable resource
Falasca and Zobel [40]	Non-expendable	–	Deterministic	To minimise cost of task shortage, total number of undesired tasks and time blocks	Exact method	– Sequencing and scheduling of volunteers are ignored
Lee et al. [3]	Both	Known	Deterministic	To minimise total weighted tardiness	Heuristic method	– Assumes sequence of visit to demand points is known in advance – Difference in quantity of non-expendable resources to be delivered is ignored
Wex et al. [15]	Non-expendable	Unknown	Deterministic	To minimise total weighted completion times	Heuristic method	– Assumes identical unit and ignores the different types and quantities of non-expendable resources to be delivered
Schryen et al. [42]	Non-expendable	Unknown	Stochastic	To minimise total weighted completion times	Heuristic method	– Model is non-linear programming – Assumes difference in quantity of non-expendable resources is negligible
Lei et al. [1]	Both	Known	Deterministic	To minimise total weighted tardiness	Heuristic method	– Model is non-linear programming – Assumes sequence of visit to demand points is known in advance
Caunhye et al. [47]	Expendable	Unknown	Stochastic	To minimise cost of setting up warehouse, total response time	Exact method	– Difference in quantity of non-expendable resources to be delivered is ignored – Only one type of expendable resource is considered
Shahparvari and Bodaghi [44]	Both	Known	Deterministic	To minimise operational costs of the vehicle fleet	Heuristic method	– Response time is calculated based on transportation time – Assumes sequence of visit to demand points is known in advance
Bodaghi et al. [48]	Both	Unknown	Probabilistic	To minimise weighted completion time	Exact method	– Uses fuzzification to generate inexact variable values, such as time windows and risk levels – The model could be applied for small to medium size case studies

fairness of human sufferings mitigation and outperform other models both in operational cost control and psychological sufferings alleviation.

While the above reviewed studies are successful in suggesting new approaches or formulating advanced strategies in solving routing and scheduling problems for emergency relief, one common shortcoming is that only single resource is considered or resource is not the focus of the optimisation. Reviews by Caunhye et al. [30]; Galindo and Batta [31]; Özdamar and Ertem [32] and Zhou et al. [33] reveal that most of the models developed for emergency relief only permit scheduling of one type of resource, either expendable or non-expendable. For example, optimisation models that consider only the resource scheduling of expendable resources include Haghani and Oh [34]; Özdamar et al. [35]; Chang et al. [36]; Balcik et al. [37]; Huang et al. [38]; Zokaee et al. [39]; among others. Another set of optimisation models consider only the resource scheduling of non-expendable resources in disaster emergencies. They include Rolland et al. [6]; Falasca and Zobel [40]; Wex et al. [15]; Lassiter et al. [41]; Schryen et al. [42]; Bodaghi and Palaneeswaran [43]; and so on.

To date, only a few studies, including Lee et al. [3]; Lei et al. [1] and Shahparvari and Bodaghi [44] have proposed models for emergency resource scheduling that include both expendable and non-expendable resources. Most of these studies put forward a mixed integer programming model (MILP) and applied heuristic algorithms for the assignment and scheduling of multiple resources in a supply network. Table 1 summarises the characteristics and limitations of some of the above mentioned vehicle routing and scheduling models for emergency relief distribution. It can be seen that many of them are deterministic in nature with minimisation of travel time as one of the objectives. Also, many of them assume that the sequence of the route is known beforehand and use heuristic methods to find a near-optimal rather than an exact solution. This is due to the NP-hard nature of the problem and it is especially so when delivery of both expendable and non-expendable resources in the

same trip is considered. In practice, this assumption may not be valid and needs to be addressed in the optimisation model.

Vehicle routing problem (VRP) for emergency relief differs from other VRPs in that vehicles are often not required to return to the depots where they are dispatched (Open VRP (OVRP)). Or if they do, the final leg is not critical. Despite there being much less research when compared to classical VRP, studies have been undertaken on OVRP using various heuristics, such as evolutionary algorithm [49], particle swarm optimisation methodology [50], genetic algorithm [51], variable neighbourhood search algorithm [52], simulated annealing [53], and ant colony optimisation [54], among others, to generate near-optimal solutions. While OVRP can have one or more starting points, in practice, multiple depots are more common to accommodate a large fleet and the class of problems is referred to as multi-depot OVRP (MDOVRP). Research on MDOVRP is relatively limited because it is more challenging than the conventional MDVRP, which is a complex combinatorial problem. For example, Tarantilis and Kiranoudis [55] studied MDOVRP using the fresh meat industry in Greece as a case study. Liu et al. [56] proposed a MILP formulation for MDOVRP and used a hybrid genetic algorithm to find routes that minimise total travelling costs. Pichka et al. [57] also proposed a MILP model for the MDOVRP and developed a simulated annealing algorithm to establish near-optimal solutions for medium-sized and large instances. For the ERO, MDOVRP is akin to the MRSRP that is investigated in this present analysis as vehicles are sent from multiple RCs; they visit more than one demand point to deliver multiple resources.

As far as heuristic methods for solving VRP for emergency response are concerned, Wex et al. [15] have systematically categorised the different types of heuristic methods and compared computationally their performance. The different heuristic methods were used to solve a rescue unit assignment scheduling problem under randomly generated scenarios using data generated from the major earthquake in Japan in

**Table 2**  
Heuristic methods for solving VRP in emergency relief proposed in recent studies.

Study	Algorithm	Purpose	Approach	Merit
Duque et al. [58]	Iterated Greedy-randomise constructive procedure (IGRCP)	For scheduling and routing of a repair crew after a disaster	Based on the GRASP meta-heuristic method with multiple runs of the construction phase plus improvement routine	Overcomes the problem size limitation of dynamic programming and solves medium- to large-scale instances efficiently
Fontem et al. [59]	Decomposition-based method	To solve the Emergency Open Routing under Stochastic Travel Times and Deadlines (EORSTTD) Problem for quick relief during emergency	Renders the EORSTTD problem tractable by formulating a counterpart problem, and decomposes it into two sub-problems	Produces a solution that enables flexible decisions to be made according to the decision-maker's preference to avoid the risk of deadline violation
Osman and Ram [60]	Centralised Point-to-Point Look-Back (C-PTPLB)	To find evacuation routes from an urban building and out of its predetermined neighbourhood	Based on looking back from intermediate destination nodes at a current time T, and identifying the objects that can be point-to-point routed to reach there precisely at time T from preceding nodes	Provides point-to-point optimal route schedules while minimising the number of iterations when compared with other methods
Bruni et al. [19]	Iterated Greedy method	For routing of vehicles carrying critical supplies and to disaster victims	Implements an adaptive local search procedure and a destroy procedure to enable extensive searching for a solution space where near-optimal solutions can be employed	Flexible and applicable to various risk measures, can provide good solutions quickly
Faiz et al. [61]	Column generation and Path generation algorithm	For vehicle routing operations during a humanitarian crisis	Devises a task adjacency graph for a path-based integer linear program, using a column generation framework to solve large-scale instances	Outperforms the exact method (traditional arc-based mixed integer linear program) in solution time without sacrificing solution quality
Moreno et al. [62]	Branch-and-Benders-cut, construction and local search heuristics	To solve the Crew Scheduling and Routing Problem in road restoration after disasters	Decomposes an integrated problem into a master problem with scheduling decisions and sub-problems with routing decisions	Provides feasible solutions and optimality gaps where instances cannot be solved utilising exact methods

2011. The heuristics examined include the Greedy method, construction heuristics, improvement heuristics, the Greedy randomised adaptive search procedure (GRASP), and the Monte Carlo-based heuristic method. Upon comparison, Wex et al. [15] reported that performance of the examined heuristic methods varies according to the size or context of the problem. Nevertheless, they concluded that using appropriate heuristic methods, reasonably feasible solutions for NP-hard routing and scheduling problem for emergency response could be obtained quickly. Table 2 summarises some of the other heuristics methods proposed by recent studies for solving the routing and scheduling problem in emergency relief distribution.

As such, an optimisation model with appropriate heuristic algorithms could serve as an automated decision support system for decision-makers in emergency situations characterised by a high level of complexity and high time pressure. They could enhance human-based decision-making through rapid scenario analysis to maximise operational efficiency in emergency situations. Doing so will minimise losses in human lives. The MRRSP model and the heuristic algorithms proposed in this study are developed with reference to the above analyses and with such an objective in mind. It aims to supplement the inadequacies of the previous studies by considering multiple resources in the problem formulation. Using mixed integer programming and heuristics to find near-optimal solution like many of the previous studies, the study extends the previous work by accommodating in the model conditions closer to reality. It not only addresses inadequacies in the extant literature of multi-resource scheduling problem for EROs but also assists decision-makers in making better decisions during emergency situations. Based on an extensive literature review, six heuristic algorithms are developed and evaluated in this study, seeking the most appropriate method for solving the MRRSP in the context of supplying both expendable and non-expendable resources from relief centres to hospitals during emergency situations. The search for most efficient heuristics for solving the MRRSP aligns with the views many researchers in this field. For example, Bélanger et al. [63] review studies on the development of models to support decision making in emergency medical services (EMS) in the last decade. The need arises from the fact that dynamism inherent to EMS, as well as considerations of equity and patient medical outcomes, have demanded for new modelling approaches to address problems related to emergency vehicle location and

relocation, in addition to dispatching decisions. While it is found that the size of the EMS problems under study is growing, and that there is an increasing use of stochastic and dynamic programming, they also opine that considerable efforts will need to be devoted to the development of more efficient solving methods.

Examples of previous efforts in this regard include Knight et al. [64]; which proposed new models for locating emergency medical services by incorporating survival functions for capturing multiple classes of heterogeneous patients, and Shahparvari et al. [18]; which developed a capacitated vehicle routing solution to evacuate short-notice evacuees with time windows and disruption risks under uncertainties during a bushfire. Bodaghi and Palaneeswaran [43]; Bodaghi et al. [4] have developed a bi-objective MILP model to address the problem where both expendable and non-expendable resources are considered. However, they did not consider the collaboration and quantity of different non-expendable resources. Wex et al. [15] have systematically categorised the different types of heuristic methods and compared computationally their performance for only the non-expendable resources. Also, they have not considered the collaboration between non-expendable resources. Table 3 gives an overview of the heuristic algorithms developed and evaluated in this study.

### 3. Notations and formulation

The notations employed in the model are as follows:

#### Sets and Indices

$V$	Set of vehicles
$I$	Set of $n$ demand points + $\{0, n+1\}$ where 0 and $n+1$ represent the dummy starting and ending nodes.
$K$	Set of relief centres (RCs)
$G$	Set of all resource types
$T$	Set of resource types for expendable resources
$R$	Set of resource types for non-expendable resources
$i, j, l$	Indices for demand points, $i, j, l \in \{0, 1, \dots, n, n+1\}$
$v$	Index for vehicle, $v \in \{1, \dots, V\}$
$k$	Index for RCs, $k \in \{1, \dots, K\}$
$t$	Index of resource types, $t \in G$



**Table 3**  
Overview of the heuristics developed in this study for solving the MRRSP for emergency response.

Heuristic	Description	Related Works	Shortcomings
<i>Greedy Algorithm</i>	<ul style="list-style-type: none"> <li>Decision rule is used to prioritise demand points (nodes) to visit</li> <li>Optimal nodes are then selected to constitute the route or schedule of the vehicle</li> </ul>	Liu et al. [65]; Ceselli et al. [66]; Majzoubi [67]; Tang and Zhu [68]; Zhao et al. [69]; Ciancio et al. [70]	<ul style="list-style-type: none"> <li>Solution may not be the best especially when there are multiple factors to consider</li> <li>Can be trapped in a local optimum instead of finding the global optimum</li> </ul>
<i>Augmented Greedy Algorithm</i>	<ul style="list-style-type: none"> <li>Adjustments are made to the priorities of the nodes based on the outcome of the Greedy algorithm</li> <li>Rerouting or rescheduling is then initiated</li> </ul>	Li and Wang [71]; Almutairi [72]; Bettinelli et al. [73]; Kritikos and Ioannou [74]	<ul style="list-style-type: none"> <li>Same issues with the Greedy algorithm except that the final solutions might be improved</li> </ul>
<i>k-Node Crossover Algorithm</i>	<ul style="list-style-type: none"> <li>Crossover procedure is applied to improve initial solution of a heuristic</li> <li>A certain number of nodes in the route are randomly exchanged prior to recalculation</li> </ul>	Baptista and Tavares [75]; Zhang et al. [76]; Zheng et al. [77]	<ul style="list-style-type: none"> <li>Longer running time due to many iterations during the crossover</li> </ul>
<i>Scheduling Algorithm</i>	<ul style="list-style-type: none"> <li>More sophisticated decision rule taking multiple factors into account is used to prioritise nodes to visit</li> <li>Adjustments are made to the priorities of the nodes based on the outcome</li> </ul>	Jaw et al. [78]; Weng et al. [79]; Ramchurn et al. [80]; Wex et al. [15]	<ul style="list-style-type: none"> <li>Can be trapped in a local optimum instead of finding the global optimum</li> </ul>
<i>Monte Carlo Algorithm</i>	<ul style="list-style-type: none"> <li>Once the nodes are prioritised, a certain percentage of them are randomised to generate a route or schedule</li> <li>Instead of a single iteration, multiple iterations are used to find near-optimal solutions</li> </ul>	Wex et al. [81]; Abdullah et al. [82]; Al-Harthei et al. [83]; Wu and Siohansani [84]	<ul style="list-style-type: none"> <li>Can be challenging to determine the appropriate level of randomness and number of iterations</li> <li>Running time can be long due to many iterations</li> </ul>
<i>Genetic Algorithm</i>	<ul style="list-style-type: none"> <li>A meta-heuristic served as a benchmark for assessing the performance of other heuristics</li> <li>Based on the principle of evolution with crossover of chromosomes, representing a sequence of nodes in a route, to find better solutions</li> </ul>	Baker and Ayechev [85]; Okhrin and Richter [86]; Zidi et al. [87]; Mguis et al. [88]; Zheng et al. [77]; Qin et al. [89]	<ul style="list-style-type: none"> <li>Running time can be long due to many iterations</li> </ul>
<i>Clustering Algorithm</i>	<ul style="list-style-type: none"> <li>A large problem is first broken down into a number of sub-problems, each with many clusters, and solved using the exact method or heuristics</li> <li>Solutions for the sub-problems are then aggregated to form the overall solution of the bigger problem</li> </ul>	Özdamar and Demir [90]; He et al. [91]; Vargas-Florez et al. [92]; Pillac et al. [17]; Gharib et al. [93]; Penna et al. [94]	<ul style="list-style-type: none"> <li>Can be challenging in splitting the original problem into an appropriate number of clusters to obtain optimality</li> <li>Running time can be long due to many iterations</li> </ul>

**Parameters**

$d_i^t$	Quantity of a resource type $t$ required by a demand point $i$
$cap_v^r$	Capacity of vehicle $v$ for carrying expendable and/or non-expendable resources
$cap_k^r$	Capacity of RC $k$ for expendable and/or non-expendable resources
$tr_{ij}^v$	Travel time required for vehicle $v$ to transport resources from $i$ to $j$
$np_i^t$	Processing time for recovery operations at demand point $i$ for a non-expendable resource $t \in R$
$cp_i^t$	Reduction factor to adjust the normal processing time of non-expendable resource $t$ per extra resources at $i$
$w_i$	Severity level of demand point $i$
$m$	An arbitrary big positive number

**Variables**

The decision variables can be defined as shown below:

$X_{ij}^v \in \{0, 1\}$	A binary variable to identify the sequence of demand points served by vehicle $v$ . $X_{ij}^v = 1$ if demand point $i$ is processed by vehicle $v$ immediately before processing demand point $j$ ; 0 otherwise.
$Y_{ki}^v \in \{0, 1\}$	A binary variable to identify an RC that can supply resources for demand point $i$ with vehicle $v$ . $Y_{ki}^v = 1$ if the RC $k$ supplies resources for demand point $i$ with vehicle $v$ ; 0 otherwise.

The auxiliary variables are:

$Q_{kv}^t$	The quantity of resource type $t$ shipped from the RC $k$ with vehicle $v$ ; $\forall k \in RC, v \in V$ .
$S_i$	Starting time of relief operation at demand point $i$ .
$C_i$	Completion time of relief operation at demand point $i$ .
$P_i^t$	Required processing time for each non-expendable resource $t$ at demand point $i$ .

With these notations, parameters, and variables, the objective function of the MRRSP model [48] is expressed as:

$$\text{Min } Z = \sum_{i=0}^n w_i C_i \tag{1}$$

The objective function (1) minimises the weighted sum of completion times over all demand points. The weighted factor depends on the severity level of each demand point. The set of constraints considered in the MRRSP model is listed below:

$$\sum_{j=1}^{n+1} \sum_{v=1}^{|V|} X_{ij}^v = 1 \quad \forall i \in I \setminus \{n+1\} \quad i \neq j \tag{2}$$

$$\sum_{j=1}^{n+1} X_{0(j)}^v = 1 \quad \forall v \in V \tag{3}$$

$$\sum_{v=1}^{|V|} X_{ij}^v + \sum_{v=1}^{|V|} X_{ji}^v \leq 1 \quad \forall i \in I \ \& \ j \in I; \quad i \neq j \tag{4}$$

$$\sum_{i=0}^n X_{il}^v - \sum_{j=1}^{n+1} X_{lj}^v = 0 \quad \forall v \in V; \forall l \in I - \{0, n+1\} \tag{5}$$

$$\sum_{k=1}^{|K|} \sum_{v=1}^{|V|} X_{ij}^v Y_{ki}^v = 1 \quad \forall i \in I - \{0, n+1\} \tag{6}$$

$$Q_{kv}^t = \sum_{i=1}^{n+1} Y_{ki}^v d_i^t \quad \forall k \in K; \forall v \in V; \forall t \in T \tag{7}$$

$$Q_{kv}^t \geq \text{Max}_i (Y_{ki}^v d_i^t) \quad \forall k \in K; \forall v \in V; \forall i \in I - \{0, n+1\}; \forall t \in R \tag{8}$$

$$\sum_{k=1}^{|K|} Y_{ki}^v \leq m \sum_{j=1}^{n+1} X_{ij}^v \quad \forall i \in I - \{0, n+1\}; \forall v \in V \tag{9}$$

$$\sum_{k=1}^{|K|} Q_{kv}^t \leq cap_v^t \quad \forall v \in V; \forall t \in G \tag{10}$$

$$P_i^t = np_i^t - cp_i^t \left( \sum_{k=1}^{|K|} \sum_{v=1}^{|V|} Q_{kv}^t Y_{ki}^v - d_i^t \right) \quad \forall i \in I - \{0, n + 1\}; \forall t \in R \quad (11)$$

$$tr_{ki}^v Y_{ki}^v \leq S_i \quad \forall v \in V; \forall i \in I - \{0, n + 1\}; \forall k \in K \quad (12)$$

$$S_i + P_i^t + tr_{ij}^v \leq S_j + m(1 - X_{ij}^v) \quad \forall i \in I - \{n + 1\} \& j \in I - \{0\}; \forall t \in R; \forall v \in V; i \neq j \quad (13)$$

$$C_i \geq S_i + P_i^t \quad \forall i \in I; \forall t \in R \quad (14)$$

$$C_i \geq 0, S_i \geq 0, P_i^t \geq 0, Q_{kv}^t \geq 0 \quad \forall i \in I; \forall k \in K; \forall v \in V; \forall t \in G \quad (15)$$

$$X_{ij}^v, Y_{ki}^v \in \{0, 1\} \quad \forall i \in I \& j \in I; \forall v \in V; \forall k \in K \quad (16)$$

Constraint sets (2) to (5) relate to determining the sequence of demand points visited by each vehicle to deliver the required resources for the recovery operations. Two additional dummy points (the starting and ending points) are defined for each vehicle (denoted by '0' for the starting point and 'n + 1' for the ending point). The processing times and transportation time from or to these dummy points are equal to zero (i.e.  $P_{0i}^{t \in R} = P_{n+1}^{t \in R} = tr_{0i}^v = tr_{i(n+1)}^v = 0$ ). Constraint (2) guarantees that there is exactly one demand point instantly processed after each demand point for each vehicle to deliver resources. Constraint (3) ensures that each vehicle initially starts from its starting point (i.e. the RC in which the vehicle is located). Constraint (4) eliminates the loops in processing the demand points by a particular vehicle. Constraint (5) guarantees that if an instant predecessor is available for each demand point and each vehicle, it would be an immediate successor in this case, unless it is in fact the last demand point (demand point n + 1).

Constraint (6) establishes the relationship between a covered incident, the vehicle (v), and the RC (k) from where the vehicle carried the resources. Constraint (7) calculates the total quantity of expendable resources delivered from RC k by vehicle v. Similarly, constraint (8) calculates the maximum quantity of non-expendable resources by a vehicle.

Constraint (9) establishes the relationship between variable  $Y_{ki}^v$  and  $X_{ij}^v$ . Constraint (10) ensures that the total quantity of resources delivered by a vehicle does not exceed its capacity. Constraint (11) calculates the processing time required for each non-expendable resource to process each demand point. The non-expendable resource usage and the processing time of the relief operation at each demand point are assumed to have a linear relationship. Each demand point has associated with it a normal processing time for each type of non-expendable resource. If the quantity of a particular non-expendable resource carried by a vehicle is greater than the requirement of a specific demand point, the processing time for the demand point is adjusted by a factor  $cp_i^t$  as described in constraint (11). Since multiplication of two decision variables ( $Q_{kv}^t$  and  $Y_{ki}^v$ ) makes this constraint non-linear, an auxiliary variable ( $H_{kvi}^t = Q_{kv}^t Y_{ki}^v$ ) with a positive real value is used to linearise this constraint as follows:

$$H_{kvi}^t \leq Q_{kv}^t + m(1 - Y_{ki}^v) \quad \forall i \in I - \{0, n + 1\}; \forall k \in K; \forall v \in V; \forall t \in R \quad (12a)$$

$$H_{kvi}^t \geq Q_{kv}^t - m(1 - Y_{ki}^v) \quad \forall i \in I - \{0, n + 1\}; \forall k \in K; \forall v \in V; \forall t \in R \quad (12b)$$

$$H_{kvi}^t \leq mY_{ki}^v \quad \forall i \in I - \{0, n + 1\}; \forall k \in K; \forall v \in V; \forall t \in R \quad (12c)$$

The developed constraints can cover both possible cases. If binary variable  $Y_{ki}^v = 1$ , then  $H_{kvi}^t = Q_{kv}^t$ . In this case, constraints (12a) and (12b) imply  $H_{kvi}^t \leq Q_{kv}^t$  and  $H_{kvi}^t \geq Q_{kv}^t$ , respectively. Since m has an arbitrary big positive value, constraint (12c) has always been satisfied in this situation, and then  $H_{kvi}^t = Q_{kv}^t$ . In the second scenario, if binary

variable  $Y_{ki}^v = 0$ , then  $H_{kvi}^t = 0$ . Since m has an arbitrary big positive value, constraints (12a) and (12b) have always been satisfied. Constraint (12c) implies  $H_{kvi}^t \leq 0$  and since  $H_{kvi}^t$  has a strictly positive real value, constraint (12c) ensures that  $H_{kvi}^t = 0$ .

Constraints (12)-(13) are time-related constraints. Constraint (12) ensures that the starting time of a relief operation at each demand point is not earlier than the latest transportation of resources from RC to demand point i with vehicle v. Constraint (13) ensures that the starting time of the relief operation is not earlier than the earliest arrival time of the resources from the previous demand points. Constraint (14) calculates the completion time of the relief operation at each demand point. Constraints (15) and (16) define the domains of employed variables.

## 4. Solution approaches

If non-expendable resources are relaxed and assuming that the processing time of operations depends on vehicle rather than resources, our model falls into the category of VRP problem. Since the VRP model is NP-hard, our model's complexity is NP-hard too and an optimal solution cannot be found for large instances [95]. In addition, decision support must be provided quickly in real emergency situations [42] and an exact solution or exact solutions cannot be identified in a reasonable amount of time for our model even for small to medium cases. Hence, several heuristics are developed to provide a high-quality solution within the running time limitation in emergency situations. In this paper, six heuristics and a mixed heuristic and Clustering algorithm are constructed and compared with the genetic algorithm. The details of these heuristics are described in the subsections below.

### 4.1. Greedy algorithm

In the Greedy algorithm, we follow the problem-solving heuristic of identifying the least weighted completion time choice as a local optimal at each stage with the intent of finding a near global optimum. To reach the minimum total weighted completion time, it is obvious that demand of incidents with high severity value (weight) should be met as early as possible. To accommodate this into the algorithm, locations (e.g. incidents/demand points) are sorted by decreasing the order of their severity level ( $w_i$ ). Starting from location i with the highest severity weight, an incident is allocated to a node (RC or a visited hospital) where the node has the least total time.

In the Greedy algorithm, the decision rule for selecting the next incident is based on identifying a node with the least total time which incorporates two components: (a) transportation time from capable nodes with capable vehicles to the next incident (i), and (b) current completion time of capable vehicles. It is worth noting that in this method processing time of non-expendable resources for the next incident (i) is not considered in the decision-making process when selecting the most suitable node and vehicle to cover the demand of the next incident. However, after selecting the most suitable vehicle to cover the next incident (i), the vehicle's completion time is updated. This is conducted by considering the following elements: (a) transportation time from the most suitable node to the next incident (i), (b) current completion time of the vehicle, and (c) maximum processing time for non-expendable resources regarding incident i. Algorithm 1 describes the pseudo code for the Greedy heuristic algorithm.

The use of the Greedy algorithm in this study has its merit because at each level of recursion the size of the problem becomes smaller. The algorithm is fast, but there is the issue of 'correctness' to overcome [96,97]. It is obvious that the obtained solution from the Greedy algorithm may not be the best, since the Greedy heuristic operates based on sorting the location for incidents by curtailing the severity level of the location. It also ignores the operations' processing times in each location in the decision-making process where the next incident to cover has to be identified.

## Algorithm 1 Greedy Algorithm.

---

**Inputs** : Sets as listed in section 3

**Parameters:** Random values for parameters listed in section 3

**Outputs** :  $Z^*, L_v, C_v$

- 1 **Initialisation**
- 2 Sort locations by decreasing  $w_i$  values.
- 3  $i = 1$ ; Initialise counter for sorted location by  $w_i$ .
- 4  $C_v \leftarrow \emptyset$ ; Initialise the current completion time of each vehicle.
- 5  $L_v \leftarrow \emptyset$ ; Initialise the current sequence of each vehicle.
- 6  $curr\_cap_k^t \leftarrow$ ; Initialise the current capacity of each RC;  $| curr\_cap_k^t = cap_k^t; \forall k; \forall t$
- 7  $curr\_cap_i^v \leftarrow$ ; Initialise the current capacity of each vehicle;  $| curr\_cap_i^v = cap_i^v; \forall v; \forall t$
- 8  $N_i \leftarrow \emptyset$ ; Initialise a list of nodes which can supply expendable and non-expendable resources to  $i$ .
- 9 **for**  $i = 1$  to  $n$  **do**
- 10     select location  $i$  to be proceed;
- 11      $N_i \leftarrow$  list the capable nodes with capable vehicles;  $curr\_cap_i^v \geq d_i^t; \forall t; curr\_cap_k^t \geq d_i^t; \forall t$  &  $v$  started from  $RC_k$ .
- 12     **if**  $N_i = \emptyset$  **then**
- 13         **abort** #Infeasible#
- 14      $TT_{N_i} \leftarrow tr_{(N_i)(i)} + C_v$  (compute list of total time values: transportation time plus the current completion time of capable vehicles).
- 15     choose  $v^*$ :  $v^* \leftarrow \{v^* | argmin_{v \in v^*}(TT_{N_i})\}$  (capable vehicle with lowest total time)
- 16     update  $C_{v^*} \leftarrow C_{v^*} \cup (Min(TT_{N_i}) + Max_{t \in R}(P_i^t))$ .
- 17     update  $L_{v^*} \leftarrow L_{v^*} \cup \{i\}$ .
- 18     update  $curr\_cap_k^t \leftarrow \{curr\_cap_k^t - d_i^t | v^* \text{ started from } RC_k\}; \forall t \in T$ .
- 19     update  $curr\_cap_i^{v^*} \leftarrow curr\_cap_i^{v^*} - d_i^t; \forall t \in T$ .
- 20      $Z_i \leftarrow w_i \times C_{v^*}$ .
- 21  $Z^* \leftarrow sum(Z_i)$
- 22 **Return**  $Z^*, L_v, C_v$

---

### 4.2. Augmented Greedy

As explained earlier, one of the main contributions of this study is to highlight that the number of available non-expendable resources affects the processing time and therefore planning of EROs. For example, if a vehicle carries three doctors and five nurses as non-expendable resources, and if the next incident requires only two nurses, since the remaining three nurses can help to expedite the operations, the processing time of this incident is reduced. Therefore, this adjustment should be performed on the Greedy algorithm to obtain a more accurate schedule.

Augmented Greedy (Algorithm 2) is developed to apply this adjustment on the output of the Greedy algorithm. In this method, the sequence of visiting incidents by each vehicle (which is the output of the Greedy algorithm) is considered as an input of the Augmented Greedy heuristic. To apply this adjustment, for each vehicle and for every non-

expendable resource, the difference between the maximum number of non-expendable resources which have been used to cover all incidents (by the current vehicle) and the demand of each incident (covered by the current vehicle) is computed. Following this, the current processing time of each incident is updated by multiplying the computed value of the difference by  $cp$  which is a random reduction coefficient. It is obvious that updating the processing time of a node ( $i$ ) affects the completion time of all subsequent nodes visited after node  $i$ . Consequently, this algorithm re-calculates the completion time of all succeeding nodes after making the adjustment on any node's processing time.

The Augmented Greedy algorithm addresses the correctness issue of the Greedy algorithm and can produce better result. Owing to the need to recalculate the completion time of all succeeding nodes upon adjustment of processing time at the proceeding node in every recursion, however, the overall processing time is much longer than that of the Greedy algorithm.



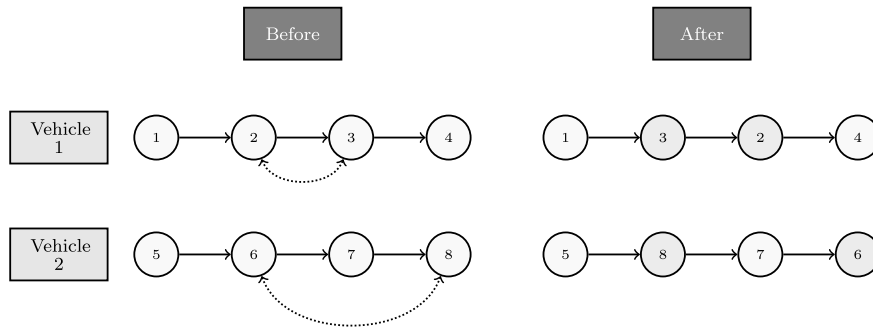


Fig. 1. k-Node Crossover algorithm.

### 4.3. k-node crossover algorithm

In the literature, Crossover procedure has been employed to improve the heuristic algorithms, where in each iteration the initial solution can be improved by applying the Crossover procedure [77]. Crossover methods can be implemented as a post-analysis improvement heuristic. In the Greedy-based algorithms it is quite possible that selecting the most optimal short-term solutions may lead to losing the chance of identifying the optimal outcome. Making the local optimal choice at each stage does not necessarily result in finding a global optimum and thus randomly exchanging the location of  $k$  number of nodes might improve the identified initial solution. After the scheduling of all

incidents for each vehicle is determined, we apply the Crossover heuristic to randomly substitute the sequence of  $k$  number of locations in each iteration.

In this study, we apply 2-node and 3-node Crossover heuristics on the output of the Augmented Greedy algorithm to further improve the scheduling of available vehicles, in order to cover the incidents (Fig. 1). In a 2-node crossover algorithm, the sequence of each vehicle's routing and its corresponding values of completion time from the Augmented Greedy algorithm are imported as inputs. Then, vehicles which have been engaged in the recovery operations are identified. Out of all engaged vehicles,  $\alpha$  per cent is randomly selected, where  $\alpha = 0.1$  in our study. If no vehicle is selected, we select at least one vehicle. Afterwards,

## Algorithm 2 Augmented Greedy Algorithm.

---

**Inputs** :  $L_v, C_v$  from Algorithm 1.  
Sets as listed in section 3.

**Parameters:** Random values for parameters listed in section 3.

**Outputs** :  $Z^*, C_v^{z^*}$

- 1 **Initialisation**
- 2  $C_v \leftarrow$ ; Import the completion time of each vehicle for all visited incidents from Algorithm 1.
- 3  $L_v \leftarrow$ ; Import the current sequence of each vehicle for all visited incidents from Algorithm 1.
- 4 **for**  $v = 1$  to  $V$  **do**
- 5     sort  $L_v$  by ascending order.
- 6      $nH_v \leftarrow |L_v|$  number of visited incidents by the current vehicle.
- 7     **if**  $nH_v \leq 1$  **then**
- 8         **go to line 4** #process next vehicle#
- 9     **for**  $s$  in  $L_v$  **do**
- 10         **for**  $t$  in  $R$  **do**
- 11             calculate  $Q_v^t$ ;  $Q_v^t \leftarrow \{Q_v^t \mid Max(D_s^t); \forall s \in L_v\}$
- 12             update  $P_s^t \leftarrow P_s^t - cp_s^t \times (Q_v^t - d_s^t)$
- 13             update  $C_v^s \leftarrow C_v^{s-1} + tr_{(s-1)(s)} + Max_t(P_s^t)$ ; (updating the completion time.)
- 14             update  $Z_i \leftarrow w_i \times C_v$ ; (updating  $Z$  for location  $i$ .)
- 15  $Z^* \leftarrow sum(Z_i)$
- 16 **Return**  $Z^*, C_v^{z^*}$

---

for each selected and engaged vehicle, the sequence of two nodes is exchanged randomly. The same process is used in the 3-node Crossover method with the exception that three nodes are randomly exchanged for each selected and engaged vehicle. Finally, the sum of weighted completion time of all vehicles is computed as the value of objective function ( $Z$ ) for the first iteration. The Crossover heuristic is iterated for a predetermined number of iterations and the minimum value of total weighted completion time of all vehicles is reported as the output of this method (Algorithm 3).

The k-Node Crossover algorithm (Algorithm 3) overcomes the shortcoming of the Greedy and the Augmented Greedy algorithms by minimising the possibilities of trapping in a local optimum. Consequently, it can give better results than the previous two heuristics. Owing to the many iterations during the crossover, however, the running time can be much longer.

#### 4.4. Scheduling algorithm

Up to this point, the processing time of the next visiting incident is

not considered in the decision to identify the least weighted completion time choice as a local optimal. To incorporate the processing times of incidents for each type of non-expendable resource in the decision-making process, a Scheduling heuristic is developed. In this algorithm, three components constitute the total time which is the basis for making decisions on the next incident to be taken care of: (a) transportation time from the current node to the next incident ( $i$ ), (b) current completion time of the vehicle at the current node, and (c) maximum processing time of non-expendable resources for incident  $i$ . The Scheduling heuristics have been previously suggested by [15,79], however, no studies have taken the multi-resource scenarios into consideration.

In this method, we introduce an improved decision rule to provide a superior optimal solution. In the Greedy algorithm, the priority of an incident is determined using its severity level ( $w_i$ ). Since processing time of the non-expendable resources has a direct impact on the completion time of a vehicle and subsequently on the objective function (total weighted completion time of all vehicles), engaging both severity level of an incident and the corresponding average processing time of non-expendable resources to define the priority of an incident seems to be

### Algorithm 3 k-Node Crossover Algorithm.

---

<b>Inputs</b>	: $L_v, C_v$ from Algorithm 2. Sets as listed in section 3. $\alpha$ : probability of selecting an engaged $v$ in the recovery operation for the k-Node Crossover process. $rep$ : number of iterations.
<b>Parameters:</b>	Same random values for parameters used in Algorithm 2.
<b>Outputs</b>	: $Z^*, C_v^{z^*}, L_v^{z^*}$
<b>1 Initialisation</b>	
<b>2</b>	$C_v \leftarrow$ ; Import the completion time of each vehicle for all visited incidents from Algorithm 2.
<b>3</b>	$L_v \leftarrow$ ; Import the current sequence of each vehicle for all visited incidents from Algorithm 2.
<b>4</b>	$V_{engaged} \leftarrow$ ; $\{V \mid  L_v  > k\}$ (vehicles which have been engaged in the recovery operations and visited more than $k$ number of incidents).
<b>5 for</b>	$r=1$ to $rep$ <b>do</b>
<b>6</b>	$N_v \leftarrow$ ; $ V_{engaged}  \times \alpha$ (number of vehicles that the crossover procedure is performed on).
<b>7</b>	<b>if</b> $N_v = 0$ <b>then</b>
<b>8</b>	$N_v = 1$
<b>9</b>	$V_{selected} \leftarrow$ ; select $N_v$ number of vehicles randomly.
<b>10</b>	<b>for</b> $v$ in $V_{selected}$ <b>do</b>
<b>11</b>	update $L_v \leftarrow \{L_v \mid k \text{ nodes are exchanged.}\}$
<b>12</b>	<b>for</b> $s$ in $L_v$ <b>do</b>
<b>13</b>	update $C_v^s \leftarrow C_v^{s-1} + tr_{(s-1)(s)} + Max_t(P_s^t)$
<b>14</b>	update $Z_i \leftarrow w_i \times C_v$ . (updating $Z$ for location $i$ .)
<b>15</b>	$Z_r \leftarrow Sum(Z_i)$
<b>16</b>	$Z^* \leftarrow Min(Z_r)$
<b>17 Return</b>	$Z^*, C_v^{z^*}, L_v^{z^*}$

---

effective. Therefore, in the Scheduling algorithm, each incident's priority is determined by increasing the order of  $\frac{\bar{P}_i}{w_i}$ , where  $\bar{P}_i$  is the average processing time of non-expendable resources, and  $w_i$  is the severity level of node  $i$ . This method addresses the trade-off between processing time and severity level for making decisions on the priority of incidents to cover.

Similar to the Augmented Greedy approach, in the Scheduling algorithm once a vehicle is allocated to cover an incident, processing time of non-expendable resources for all visited nodes by this vehicle is updated. In other words, for each non-expendable resource the maximum value of processing time that has been required for each visited incident is identified. Then, processing time of all visited nodes including the current incident is reduced by this term: a random reduction coefficient ( $cp$ ) times the difference between maximum value of demand for all visited incidents by the current vehicle and demand of each incident for non-expendable resources. Details concerning the Scheduling heuristic are explained in Algorithm 4.

The Scheduling algorithm has the merit of completing more tasks in a given amount of time by running tasks which take less time to complete. It thus can have a higher throughput rate than the previous algorithms. The downside is that tasks that take longer time to complete might have a lower opportunity to run. The search can also be trapped in a local optimum instead of finding the global optimum.

#### 4.5. Monte Carlo algorithm

The proposed Scheduling heuristic algorithm provides significant benefits that can solve parallel vehicle disaster recovery operations.

##### Algorithm 4 Scheduling Algorithm.

---

**Inputs** : Sets as listed in section 3.

**Parameters:** Random values for parameters listed in section 3.

**Outputs** :  $Z^*, L_v, C_v$

- 1 **Initialisation**
- 2 sort locations by increasing values  $\text{Min}(\frac{p_i}{w_i}) \mid \frac{p_1}{w_1} \leq \frac{p_2}{w_2} \leq \dots \leq \frac{p_n}{w_n}; p_i \leftarrow \text{Average}(p_i^t)_{t \in R}$
- 3  $i = 1$ ; Initialise counter for sorted locations.
- 4 steps 5-14 from Algorithm 1.
- 5  $TT_{N_i} \leftarrow tr_{(N_i)(i)} + C_v + \text{Max}_{t \in R}(P_i^t)$  (compute total time values: transportation time plus the current completion time of capable vehicles plus maximum processing time of non-expendable resources for  $i$ ).
- 6 choose  $v^*$ :  $v^* \leftarrow \{v^* \mid \text{argmin}_{v \in v^*}(TT_{N_i})\}$  (capable vehicle with lowest total time)
- 7 update  $C_{v^*} \leftarrow C_{v^*} \cup \text{Min}(TT_{N_i})$ .
- 8 steps 17-20 from Algorithm 1.
- 9 **for**  $s$  in  $L_{v^*}$  **do**
- 10     **for**  $t$  in  $R$  **do**
- 11         calculate  $Q_v^t$ ;  $Q_v^t \leftarrow \{Q_v^t \mid \text{Max}_t(D_s^t); \forall s \in L_{v^*}\}$
- 12         update  $P_s^t \leftarrow P_s^t - cp_s^t \times (Q_v^t - D_s^t)$
- 13         update  $C_{v^*}^s \leftarrow C_{v^*}^{s-1} + tr_{(s-1)(s)} + \text{Max}_t(P_s^t)$ ; (updating the completion time.)
- 14         update  $Z_i \leftarrow w_i \times C_{v^*}^s$ ; (updating  $Z$  for location  $i$ .)
- 15  $Z^* \leftarrow \text{Sum}(Z_i)$
- 16 **Return**  $Z^*, L_v, C_v$

---

However, due to the complexity of the problem, this method might get stuck in a local optima. To further improve this method and to overcome the informational uncertainties, we have customised the Monte Carlo method to deal with this problem. The structure of this method is similar to the Scheduling heuristic algorithm except for the decision-making process to select which vehicle is going to cover the next incident ( $i$ ). Once a list of capable nodes with capable vehicles is identified, the vehicle with the lowest total time is selected to undertake the recovery operations for location  $i$  in the Scheduling method. The difference here is that instead of selecting the vehicle with the least total time, a list of all capable vehicles are sorted based on their estimated total time, and then  $\beta\%$  of vehicles with least total time are screened. Afterwards, one vehicle is randomly selected from the screened list. By introducing randomness into the process of identifying the local optimal choice and repeating this process for a predefined number of iterations, the schedule with minimum total weighted completion time generally outperforms the Scheduling method, which seeks the near optimal solution in a single iteration. Algorithm 5 delineates the process of applying the Monte Carlo method.

The Monte Carlo algorithm has all the merits of the Scheduling algorithm but minimising the bias against tasks with long completion time. As such, it reduces the possibility of trapping in a local optimum and can produce a better solution. Nonetheless, it can be challenging to determine the appropriate level of randomness and number of iterations. The algorithm can also have a long running time due to many iterations.

#### 4.6. Meta-Heuristic algorithm (GA)

Meta-heuristic approaches are widely used as an alternative

## Algorithm 5 Monte Carlo Algorithm.

---

<b>Inputs</b>	: Sets as listed in section 3.
	$\beta$ : proportion of vehicles with the least completion time (to be selected).
	$rep$ : number of iterations.
<b>Parameters:</b>	Random values for parameters listed in section 3.
<b>Outputs</b>	: $Z^*, C_v^{z^*}, L_v^{z^*}$

---

- 1 **Initialisation**
- 2 **for**  $r=1$  to  $rep$  **do**
- 3     steps 2-5 from Algorithm 4.
- 4      $v_{set}^*$ :  $v_{set}^* \leftarrow \text{Sort}(v_{capable})$  (create a list of capable vehicles, sorted by ascending order based on estimated total time for covering incident  $i$ )
- 5     update  $v_{set}^*$ :  $v_{set}^* \leftarrow \{v \mid \text{Index}(v^*) \leq \text{Int}(\beta\% \times |v_{set}^*|); \forall v \in v_{set}^*\}$  (vehicles with top  $\beta\%$  lowest completion time)
- 6     **if**  $|v_{set}^*| = 0$  **then**
- 7         choose  $v^*$ :  $v^* \leftarrow \{v \mid \underset{v \in v_{set}^*}{\text{argmin}}(TT_{Ni})\}$
- 8         **else**
- 9             choose randomly  $v^*$ :  $v^* \leftarrow \{v \mid \forall v \in v_{set}^*\}$  (choose one vehicle randomly)
- 10     update  $C_{v^*} \leftarrow C_{v^*} \cup TT_i^{v^*}$
- 11     steps 8-14 from Algorithm 4.
- 12      $Z_r \leftarrow \text{Sum}(Z_i)$
- 13  $Z^* \leftarrow \text{Min}(Z_r)$
- 14 **Return**  $Z^*, C_v^{z^*}, L_v^{z^*}$

---

approach for heuristics in disaster relief operations [77]. In particular, a Genetic Algorithm (GA) can be designed as an alternative for proposed heuristics and compare the performance of heuristics against GA. An evolutionary optimisation algorithm, such as GA, conducts a search through the possible solutions by exploiting a population of points in parallel rather than a single point [98]. In this study, a chromosome represents the sequence of incidents that is assigned to each vehicle and it is generated randomly. As an iterative process, the evolution starts from a population of a randomly generated sequence of vehicles where the population in each iteration is referred to as a generation. In each generation, the fitness of every chromosome is evaluated based on: firstly, the total weighted completion time of all vehicles (value of objective function); and secondly, penalty values that are applied for exceeding the capacity of a corresponding RC or vehicle for both expendable and non-expendable resources.

The better fitting chromosomes are stochastically selected from the current population, and each individual's genome is modified to form a new generation. In other words, the chromosomes are sorted based on their fitness values and a proportion (i.e. survival rate) of chromosomes with the highest fitness are selected as the parents for the next generation. We have adopted the single-point crossover genetic operator in which genetic information of two parents is combined to generate new offspring. In this study, the crossover point is randomly selected from the range of incidents. To maintain genetic diversity from one generation of a population of chromosomes to the next, based on the defined mutation

probability, sequences for visiting the incidents in a chromosome are altered from its initial state. This evolutionary process is iterated for a predefined period of time.

Although the genetic algorithms do not guarantee optimality, they are likely to provide a solution that is closest to the global optimum. The search is not easily trapped in local optima because of the probabilistic nature of the solution [99]. As such, the solution obtained using genetic algorithm can be used a benchmark for the other heuristics. The downside of the algorithm is that the running time can be long due to many iterations especially for large complex problems.

### 4.7. Clustering algorithm

Finding the optimal solution using the optimisation model may not be practical in some large and highly complex cases [100]. Therefore, most heuristic approaches try to improve the initial solution based on the predefined rules and identifying the local optimal choice with the hope of achieving the global optimal solution. Alternatively, clustering approaches have been introduced to manage the complexity of a large sized problem by assigning the demand points to a fixed number of clusters and then applying the exact mathematical model for the reduced number of incidents in each cluster. However, it is still possible that the size of a cluster's sub-problem would be large enough to make it NP-Hard and unable to be solved. To overcome this issue, Ozdamar and Demir [90] proposed a hierarchical clustering and routing procedure for

large-scale disaster relief logistics planning through an iterative process. Firstly, an attempt is made to solve the main problem within a predefined period of time. If no solution is identified, then the problem is divided into two sub-problems and another run is performed. If a solution is obtained for each cluster, solutions are aggregated to constitute the overall solution to the problem. In the case where a feasible solution is lacking for both sub-problems, clusters are further broken down into smaller sub-problems until the exact model can be solved for all sub-problems. This iterative clustering approach is adopted in this study. However, there is a major issue to address: how can the incidents be clustered to achieve the maximum precision for the aggregated solution?

We have developed a novel methodology to hierarchically cluster the recovery operations where the processing time of non-expendable resources is accounted for. First, for each pair of the nodes in the sequence of  $(i-1)$  in the main problem (including RCs and incidents), a relatedness function  $(r_{(i-1)i})$  is defined by computing the ratio of the normalised severity level  $(w_i)$  over the normalised total completion time (TC) of covering incident  $i$ . Total completion time includes the transportation time from node  $i-1$  to  $i$  and maximum processing time of non-expendable resources for incident  $i$  (i.e.  $tr_{(i-1)i} + \text{Max}_{t \in R}(P_i^t)$ ). This function addresses the trade-off between the processing time of an incident and its severity level when deciding to select an appropriate vehicle that covers the next incident. Given that the starting point of all vehicles is a RC, and a RC might have more than a single vehicle, the relatedness value  $r_{ki}^v$  is computed for each vehicle  $v$  that is available in a RC and an incident  $i$ . To obtain the relatedness value for two incidents (e.g.  $i-1$  and  $i$ ), since a single vehicle is available in the current incident  $(i-1)$ , there would be no need to consider the multi-vehicle scenario and  $r_{(i-1)i}$  can be computed from the ratio of  $\left(\frac{w_i}{TC_{(i-1)i}}\right)$ .

Once the relatedness for each pair of nodes is obtained, incidents are tentatively allocated to a vehicle using a heuristic method. After ensuring that a RC ( $k$ ) and its available vehicle(s) have sufficient capacity to supply expendable and non-expendable resources for an incident, one would be nominated to cover incident  $i$  which has the maximum relatedness  $r_{ki}$  to it. This process is repeated for all available vehicles. For covering the next set of incidents, similarly, an incident  $i$  is allocated to the vehicle that is located in node  $i-1$  where  $r_{(i-1)i}$  is the highest. By using this method, total  $n$  number of incidents ( $n \in I$ ) are broken down to  $v$  number of collections ( $I_v$ ) which will be potentially covered by the corresponding vehicle ( $v$ ). Finally, the completion time ( $C_v$ ) of each vehicle visiting the related nodes  $I_v$  is computed and the weighted completion time of each node ( $Z_i$ ) is obtained. The total weighted completion time of each vehicle  $Z_i^v$  would be the sum of  $Z_i$  for all incidents that are visited by vehicle  $v$ . Aggregating the total weighted completion time of all vehicles gives the value of objective function ( $Z^*$ ).

Similar to the Augmented Greedy method, completion time of each vehicle and the weighted completion time of each incident ( $Z_i$ ) should be updated to reflect the fact that where extra non-expendable resources are available in a vehicle, they definitely assist in processing the incident faster and the completion time of incidents decreases accordingly. Updating the processing time for all nodes is performed prior to creating sub-problems and their corresponding clusters.

Considering the fact that scheduling of EROs should be carried out in a fairly short time span, and since attempting to identify a solution for a cluster is a lengthy process, we limit the maximum number of trials to the number of vehicles in a problem. Therefore, for  $v$  number of vehicles, there are  $p = v$  number of sub-problems in the most broken down sub-problem. Each sub-problem includes  $U^p$  number of clusters. In each cluster, there are  $K_{it}^p, V_{it}^p, I_{it}^p$  number of RCs, vehicles, and incident sets, respectively.

To form a larger problem which includes  $v-1$  number of clusters, two clusters should be merged. We combine the two clusters with the lowest value of total weighted completion times ( $Z_i^v$ ) following the logic that an exact model can identify a better solution when more incidents

are available in a pool. This process is repeated to form sub-problems with  $(v-1, v-2, \dots, 1)$  number of clusters. It is obvious that a problem with '1' cluster is the same as the original problem.

The process of identifying a solution starts with the problem which has a single cluster including all vehicles, in other words, the original problem. We attempt to solve this problem using the exact mathematical model for a predefined period of time to meet a target gap. If a solution is identified, the process is terminated; otherwise, the algorithm attempts to solve the sub-problem with two clusters. This process continues until either a solution is identified for all clusters of a sub-problem, or the last sub-problem including  $v$  number of clusters delivers no feasible solution. Algorithm 6 displays the detailed steps that are involved in the Clustering algorithm.

The Clustering algorithm has the advantage of finding a solution for a large complex vehicle routing and scheduling problem by splitting the problem into sub-problems of smaller clusters to solve, which is relatively easier, and combining the outcomes to form a total solution. It can provide a good balance between effort and quality of solution [101]. The shortcoming is that it can be challenging in splitting the original problem into an appropriate number of clusters to obtain optimality. Also, because of the split, running time can be long due to many iterations.

## 5. Numerical experiment

In this section, numerical experiments are developed to evaluate and compare the proposed algorithms. To reflect the impact and diversity of real-world natural disasters, ten scenarios have been generated. In the next section, settings of each experiment are explained. Next, the validity of the developed heuristics, meta-heuristic, clustering, and exact models are investigated. Finally, results of solving the developed scenarios using the proposed methods are presented.

### 5.1. Settings

Random variables and their distribution are presented in Table 4. Since generating random numbers that follow a Normal distribution may result in negative values, a sub-routine has been developed to regenerate the random numbers when a negative value exists in a set. With regard to the capacity of RCs and vehicles for expendable and non-expendable resources, Normal distribution is adopted. However, the parameters  $(\mu, \sigma)$  of this distribution for the aforementioned four random variables are selected in way that they suit the problem size. Without loss of generality, two expendable and two non-expendable resources have been considered in this study.

The special settings of individual algorithms are presented below:

### 5.2. Validation

The developed exact mathematical model, heuristics, meta-heuristic, and Clustering algorithms should be validated prior to putting them into

k-Node Crossover	A	10%
	Number of iterations ( <i>rep</i> )	10,000
Monte Carlo	B	30%
	Number of iterations ( <i>rep</i> )	10,000
GA	Run time	30 min
	Population size	1000
	Number of generations	500
	Number of chromosomes	100
	Survival rate	70%
	Mutation probability	50%
	k-penalty cost	100,000
Clustering	Run time	2 h
	Target Gap ( <i>gap<sub>t</sub></i> )	1%
	BigM	1,000,000



## Algorithm 6 Clustering Algorithm.

---

**Inputs** : Sets as listed in section 3.

$gap_t$ : target gap.

$t_e$ : period of time that exact model is run for a cluster.

**Parameters:** Random values for parameters listed in section 3.

**Outputs** :  $Z^*$ ,  $C_v^{z^*}$ ,  $L_v^{z^*}$

- 1 **Initialisation (Clustering)**
- 2  $C_v \leftarrow \emptyset$ ; Initialise the current completion time of each vehicle.
- 3  $L_v \leftarrow \emptyset$ ; Initialise the current sequence of each vehicle.
- 4  $curr\_cap_k^t \leftarrow$ ; Initialise the current capacity of each RC;  $| curr\_cap_k^t = cap_k^t; \forall k; \forall t$
- 5  $curr\_cap_t^v \leftarrow$ ; Initialise the current capacity of each vehicle;  $| curr\_cap_t^v = cap_t^v; \forall v; \forall t$
- 6 **for**  $v$  **in**  $V$  **do**
- 7 **for**  $i$  **in**  $I$  **do**
- 8 choose  $k$ :  $k \leftarrow \{k \mid v \text{ starts from } k\}$
- 9  $TC_i^v \leftarrow tr_{ki} + \underset{t \in R}{Max}(P_i^t)$  (compute completion time)
- 10  $r_{ki}^v \leftarrow \frac{Normalised(w_i)}{Normalised(TC_i^v)}$ ; or 0 if  $k$  or  $v$  has insufficient capacity. (compute the relatedness between RC  $k$  and incident  $i$ )
- 11 **for**  $i$  **in**  $I$  **do**
- 12 **for**  $j$  **in**  $I$  **do**
- 13  $TC_{ji} \leftarrow tr_{ji} + \underset{t \in R}{Max}(P_i^t)$  (compute completion time)
- 14  $r_{ji}^v \leftarrow \frac{Normalised(w_i)}{Normalised(TC_{ji})}$ ; or 0 if  $k$  or  $v$  has insufficient capacity. (compute the relatedness between incidents  $j$  and  $i$ )
- 15 **for**  $v$  **in**  $V$  **do**
- 16 **if**  $v$  **is in starting point**  $RC_k$  **then**
- 17 choose  $i$ :  $i \leftarrow \{i \mid \underset{i \in I}{argmax}(r_{ki})\}$
- 18 update  $C_v \leftarrow C_v \cup (tr_{ki} + \underset{t \in R}{Max}(P_i^t))$  (compute completion time)
- 19 update  $L_v \leftarrow L_v \cup i$
- 20 **else**
- 21 choose  $i$ :  $i \leftarrow \{i \mid \underset{i \in I}{argmax}(r_{(i-1)i})\}$
- 22 update  $C_v \leftarrow C_v \cup (C_v^{(i-1)} + tr_{(i-1)i} + \underset{t \in R}{Max}(P_i^t))$  (compute completion time)
- 23 update  $L_v \leftarrow L_v \cup i$
- 24 update  $curr\_cap_k^t \leftarrow \{curr\_cap_k^t - d_i^t \mid v \text{ started from } RC_k\}; \forall t \in T.$
- 25 update  $curr\_cap_t^v \leftarrow curr\_cap_t^v - d_i^t; \forall t \in T.$
- 26  $Z_i \leftarrow w_i \times C_v$

---

---

```

26 for  $v$  in  $V$  do
27   for  $s$  in  $L_v$  do
28     for  $t$  in  $R$  do
29       calculate  $Q_v^t$ ;  $Q_v^t \leftarrow \{Q_v^t \mid \text{Max}(D_s^t); \forall s \in L_v\}$ 
30       update  $P_s^t \leftarrow P_s^t - cp_s^t \times (Q_v^t - D_s^t)$ 
31       update  $C_v^s \leftarrow C_v^{s-1} + tr_{(s-1)(s)} + \text{Max}(P_s^t)$ ; (updating the completion time.)
32       update  $Z_i \leftarrow w_i \times C_v$ ; (updating  $Z$  for location  $i$ .)
33  $SP_p \leftarrow \emptyset$  initialise sub-problems set.
34  $U_{kvi}^p \leftarrow \emptyset$  initialise  $u$  clusters for sub-problem  $p$  which includes  $k$ ,  $v$ ,  $i$  number of RCs, vehicles, and incidents,
    respectively.
35  $K_u^p \leftarrow \emptyset$  initialise  $k$  RCs for cluster  $u$  of sub-problem  $p$ .
36  $V_u^p \leftarrow \emptyset$  initialise  $v$  vehicles for cluster  $u$  of sub-problem  $p$ .
37  $I_u^p \leftarrow \emptyset$  initialise  $i$  incidents for cluster  $u$  of sub-problem  $p$ .
38  $Z_u^p \leftarrow \emptyset$  initialise total weighted completion time for cluster  $u$  of sub-problem  $p$ .
39 for  $p=|V|$  to 1 do
40   for  $u=1$  to  $p$  do
41     if  $p=|V|$  then
42        $V_u^p \leftarrow v_u$ 
43     else
44        $V_1^{(p-1)}, V_2^{(p-1)} \leftarrow \{V_u^{(p-1)} \mid \underset{V_u^{(p-1)}}{\text{argmin}}(C_u^{(p-1)})\}$  (select two vehicle sets from previous sub-problem
        with the lowest completion time)
45        $V_1^p \leftarrow V_1^{(p-1)} \cup V_2^{(p-1)}$ 
46        $V_u^p \leftarrow \{V_u^{(p-1)} \mid V_u^{(p-1)} \notin \{V_1^{(p-1)}, V_2^{(p-1)}\}\}$ 
47        $K_u^p \leftarrow \{k \mid k \in K; V_u^p \text{ starts from } k\}$ 
48        $I_u^p \leftarrow \{i \mid i \in I; i \text{ is allocated to } V_u^p\}$ 
49        $Z_u^p \leftarrow \sum_{i \in I_u^p} (Z_i)$ 
50    $Z^p \leftarrow \sum_{u \in U^p} (Z_u^p)$ 

```

---

```

51 Initialisation (Solving)
52 for  $p=1$  to  $|V|$  do
53     for  $u=1$  to  $p$  do
54         while  $time < t_e \wedge gap > gap_t$  do
55             Run exact model on cluster  $u$  of sub-problem  $p$ .
56         if  $gap > gap_t$  then
57             goto line 52.
58         else
59              $Z_u^p \leftarrow Z^*$ 
60     if  $Z_u^p \neq \emptyset \forall u$  then
61          $Z^p \leftarrow \sum_{u \in U^p} (Z_u^p)$ 
62          $Z^* = Z^p$ 
63     abort #solution found#
64 if  $Z^* = \emptyset$  then
65     abort #no solution found#
66
67 Return  $Z^*, C_v^{z^*}, L_v^{z^*}$ 

```

practice. Four small sized scenarios are considered for the validation process (Table 5): (1,2,1), (1,4,2), (2,5,3), and (2,8,4) in the format of (number of RCs, number of incidents, and number of vehicles). Python 3.6 is employed for evaluating the heuristics, meta-heuristic, and some parts of the Clustering algorithm. To program and solve the exact model, CPLEX 12.8 is used in conjunction with Python 3.6 API.

Results of the validation process indicate a satisfactory outcome and show consistent solutions for the developed methods. Greedy algorithm is not comparable with others as the processing time of a node is not adjusted for the cases where extra non-expendable resources are available to expedite the processing of an incident. However, the smallest scenario shows that when processing time of nodes is updated based on available non-expendable resources in a vehicle (i.e. Augmented Greedy), the total weighted completion time would be the same for all methods. The exception here is the 3-node crossover algorithm which did not provide a result. It makes sense as there are only two incidents in this scenario and applying the 3-node crossover is not possible. We have also manually solved the first two scenarios to ensure that the results of algorithms are valid. The outcomes of the remaining experiments demonstrate that the developed models and methods are valid.

**Table 4**  
Random variables and their distribution.

Random Variable	Distribution	Random Variable	Distribution
Processing time $N_{ex_1}$	$\sim \mathcal{N}(50, 15)$	Demand $Ex_1$	$\sim \mathcal{U}(20, 50)$
Processing time $N_{ex_2}$	$\sim \mathcal{N}(50, 15)$	Demand $Ex_2$	$\sim \mathcal{U}(20, 50)$
Travel time	$\sim \mathcal{N}(5, 1)$	Demand $N_{ex_1}$	$\sim \mathcal{U}(1, 10)$
Severity Level (Weight)	$\sim \mathcal{U}(1, 6)$	Demand $N_{ex_2}$	$\sim \mathcal{U}(1, 10)$
CP $N_{ex_1}$	$\sim \mathcal{N}(0.9, 0.02)$	CP $N_{ex_2}$	$\sim \mathcal{N}(0.9, 0.02)$

### 5.3. Results

**Table 5**  
Assessing the validity of the exact, heuristic, and meta-heuristic models.

Scenario	1	2	3	4
No RCs	1	1	2	2
No Incidents	2	4	5	8
No Vehicles	1	2	3	4
Greedy	798	1883	1381	1770
Augmented Greedy	790	1823	1365	1714
2-Node Crossover	790	1823	1365	1696
3-Node Crossover	-	1823	1365	1714
Scheduling	790	1766	1323	1696
Monte Carlo	790	1766	1323	1668
GA	790	1766	1323	1668
Clustering	790	1766	1323	1668
Exact	790	1766	1323	1668

To assess effectiveness and accuracy of the developed methods for responding to the EROs, ten scenarios have been developed. To ensure that variations of the model inputs are captured in the objective function, each scenario is analysed under all heuristics for thirty iterations. The mean and standard deviation of thirty runs for all heuristics are presented in Table 6. The only exception is the Clustering method which is evaluated once. Compared to other developed methods, computational time of Clustering method is so high that its usability for EROs is questionable. Results indicate that lack of updating the processing time (i.e. Greedy algorithm) where extra non-expendable resources are available, leads to ineffective planning of EROs. Furthermore, the k-Node Crossover algorithm generally makes some improvements to the

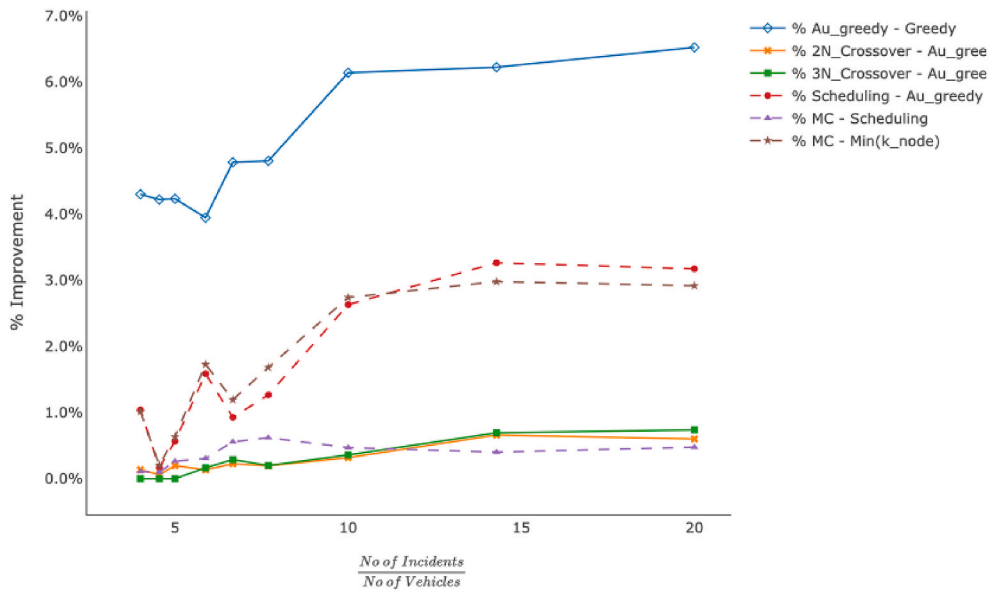


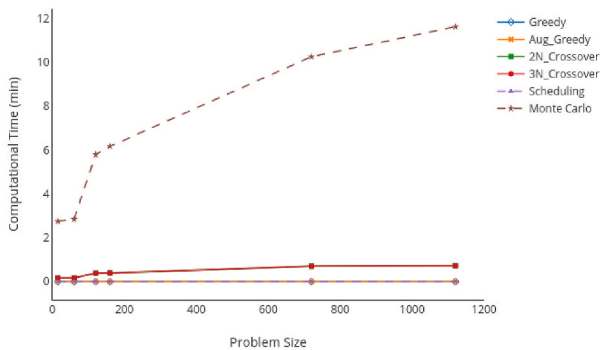
Fig. 2. Percentage of improvement for developed heuristics.

output of the Augmented Greedy method. It can be observed that incorporating the updated processing time when deciding to select the next incident (i.e. Scheduling algorithm) is definitely effective. This algorithm is further improved by randomly selecting a capable vehicle among the candidates with the least total completion time (i.e. Monte Carlo method), in order to avoid trapping in a local optimum. Finally, neither GA nor Clustering demonstrates outstanding performance in planning of recovery operations.

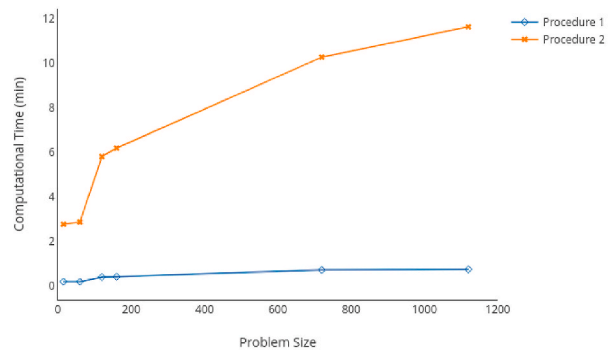
Considering the methods which have been used for developing the heuristics, the number of allocated incidents to each vehicle might affect the model's effectiveness. To further investigate this proposition, percentage of improvement for each pair of related heuristics against the ratio of number of incidents to number of vehicles for various problems is presented in Fig. 2. When this ratio increases, it implies that vehicles visit a larger number of incidents in a given scenario. Results indicate that if more incidents are allocated to a vehicle, the Augmented Greedy algorithm can deliver a better solution. The logic behind this finding is that the process of adjusting the processing time for an incident in this method impacts on the completion time of all subsequent incidents which are visited by the same vehicle. Therefore, a vehicle visiting more incidents usually improves the total completion time so that all allocated

incidents are in fact covered.

The same pattern can be observed for both 2-Node and 3-Node Crossover methods. If more incidents are allocated to a vehicle, the k-Node Crossover algorithm has further opportunity to identify a better solution by exchanging the sequence of the higher number of visited incidents. It can also be observed that 3-Node method outperforms the 2-Node Crossover when more incidents are visited by a vehicle. Higher values of this ratio also lead to higher performance of the Scheduling algorithm compared to the Augmented Greedy. This is due to the fact that in the Scheduling algorithm, processing time of the next incident is involved in the decision making on identifying the next most suitable incident to cover. The improved decision making process has a positive impact on the total completion time as the required time to complete all incidents is reduced accordingly. The same logic applies when the magnitude of improvement regarding the Monte Carlo algorithm is assessed against the minimum improvement of k-Node Crossover algorithms. Compared to the Scheduling algorithm, the Monte Carlo post-analysis algorithm also works relatively better in scenarios that number of incidents is higher and thus more incidents are allocated to vehicles. However, this method is almost robust when changes in this ratio occur.



(a)



(b)

Fig. 3. Comparison between the computational time of (a) heuristics, and (b) Procedure 1 (Greedy + Augmented Greedy + Min(K-Node Crossover)) and Procedure 2 (Scheduling + Monte Carlo).

**Table 6**  
Results of solving ten scenarios using the developed heuristics, GA, and Clustering algorithms.

Scenario		1	2	3	4	5	6	7	8	9	10
No RCs		2	2	3	4	4	5	6	7	10	15
No Incidents		20	20	40	40	60	60	100	100	500	500
No Vehicles		3	4	6	7	9	12	18	20	30	50
Greedy	$\mu$	13,509	10,506	26,899	23,659	40,426	31,984	57,537	53,955	750,899	470,046
	$\sigma$	1,462	1,398	2,363	2,023	3,011	2,283	3,713	3,377	19,978	10,125
Aug Greedy	$\mu$	12,837	10,059	25,681	22,562	38,582	30,621	55,022	51,714	710,737	447,091
	$\sigma$	1,401	1,357	2,316	1,932	2,887	2,212	3,586	3,274	19,143	9,729
2-N Crossover	$\mu$	12,723	10,000	25,529	22,469	38,433	30,521	54,914	51,564	709,572	446,058
	$\sigma$	1,363	1,344	2,302	1,921	2,878	2,198	3,577	3,242	19,555	9,708
3-N Crossover	$\mu$	12,718	10,025	25,524	22,476	38,426	30,527	54,913	51,644	708,710	446,778
	$\sigma$	1,334	1,326	2,289	1,914	2,876	2,178	3,576	3,213	19,554	9,781
Scheduling	$\mu$	12,592	9,949	25,221	22,278	37,984	30,247	54,266	50,986	696,016	439,345
	$\sigma$	1,338	1,309	2,187	1,908	2,856	2,124	3,446	3,049	18,849	9,256
Monte Carlo	$\mu$	12,592	9,809	24,999	22,029	37,654	30,013	54,129	50,968	694,502	437,111
	$\sigma$	1,338	1,315	2,194	1,887	2,839	2,095	3,466	3,036	18,622	9,454
GA	$\mu$	12,978	10,138	27,470	24,205	42,712	33,757	64,112	59,873	910,870	582,565
	$\sigma$	1,302	1,294	2,292	1,994	2,881	2,178	3,522	3,032	22,186	11,710
Clustering	$\mu$	14,284	13,148	28,738	22,147	45,684	32,250	60,124	51,531	NS	NS

Given that this study investigates the decision support systems for ERO, probing the computational time for obtaining a near-optimal schedule is crucial. Fig. 3 presents the mean computational time of heuristic-based methods for thirty runs. Mean and 95% confidence interval of all runs are presented in A. It is observed that all base methods including Greedy, Augmented Greedy, and Scheduling algorithms are able to solve the examined scenarios using two Intel Xeon processors (2.5 GHz) with 16 GB of RAM in less than a minute. However, the post-analysis algorithms including k-Node Crossover and Monte Carlo methods where the schedule is being revisited for 10,000 iterations trigger the significant increase in computational time (Fig. 3a). Therefore, we have bundled the computational time of each base method with its post-analysis improvement algorithms in two procedures.

The first procedure includes the sum of computational times of Greedy, Augmented Greedy, and minimum computational time of 2-Node and 3-Node Crossover methods. The second procedure includes the sum of computational times of Scheduling and Monte Carlo algorithms. Although section 5.3 has shown that the second procedure is superior in terms of identifying the least total weighted completion time, Fig. 3b illustrates that the computational time of this procedure is more sensitive to the size of the problem (i.e. number of incidents  $\times$  number of RCs  $\times$  number of vehicles). Scheduling, however, can be performed without engaging the Monte Carlo as the post-optimal improvement algorithm, and thus Scheduling provides the best of two worlds in delivering a reasonable solution with a fair amount of computational time. It is worth noting that clustering has not been considered for this analysis as the computational time that is required for this method has a very wide range, starting from 30 min for smaller problems to more than two days for larger problems in the range of examined scenarios.

#### 5.4. Case study

To enable the validation of the proposed model, a case study of multi-resource scheduling for emergency operations in the public hospital

**Table 7**  
Results of the comparison of the algorithms on the real case study.

Algorithms	Greedy	Updated Greedy	2-N Crossover	3-N Crossover	Scheduling	Monte Carlo	GA	Clustering
Objectives	935	775	775	775	772	757	757	838
CPU times(s)	0.10	0.08	11.64	10.69	0.10	187.92	1800	7651

network in the Melbourne metropolitan area was brought forth for succinct analysis. The case study assumed a sudden outbreak scenario similar to the 2003 SARS coronavirus outbreak in Asia and other parts of the world Bodaghi et al. [4]. The Information regarding the hospitals in the Melbourne metropolitan area was obtained from the Department of Health, Victoria, Australia and the National Health Performance Authority [102]. The validation exercise included 17 hospitals (demand points) as listed in Table A2 and portrayed in Fig. 4. Furthermore, the case study evaluation assumed two relief centres (RCs) for distributing resources. The stipulated RCs being Melbourne’s Tullamarine Airport and Moorabbin Airport while transportation was by road for both expendable resources (e.g., medical supplies) and non-expendable resources (e.g., surgeons). The destinations were the emergency operations demand points i.e. hospitals. The simulated case study was deemed to be equivalent to emergency operations in a disaster relief scenario.

In addition, it was assumed that four capable vehicles were available. Two vehicle types, namely heavy and light emergency vehicles with specific capacities, were included in this validation, i.e. light vehicle 1 and heavy vehicle 2 were available in Tullamarine Airport, and light vehicle 3 and heavy vehicle 4 were available in Moorabbin Airport. The release times of vehicles from Tullamarine Airport and Moorabbin Airport were randomly generated using a uniform distribution  $U(0, 2)$ . The shortest travel time between nodes with an average speed of 40 km/h, was the basis for calculating resource transportation for emergency operations at the demand points.

In this case study, two types of non-expendable resources were considered: (a) health professionals (excluding surgeons) for non-surgery emergencies, and (b) surgeons for emergency surgery. The numbers of health professionals and surgeons available at the Tullamarine and Moorabbin Airports were considered (60, 35) and (40, 20), respectively. Alternatively, the modelling process made assumptions regarding the treatments undertaken by each health professional teams. In this sense, each team was stipulated to offer treatment to 40 emergency patients and three resuscitation patients. The two types of



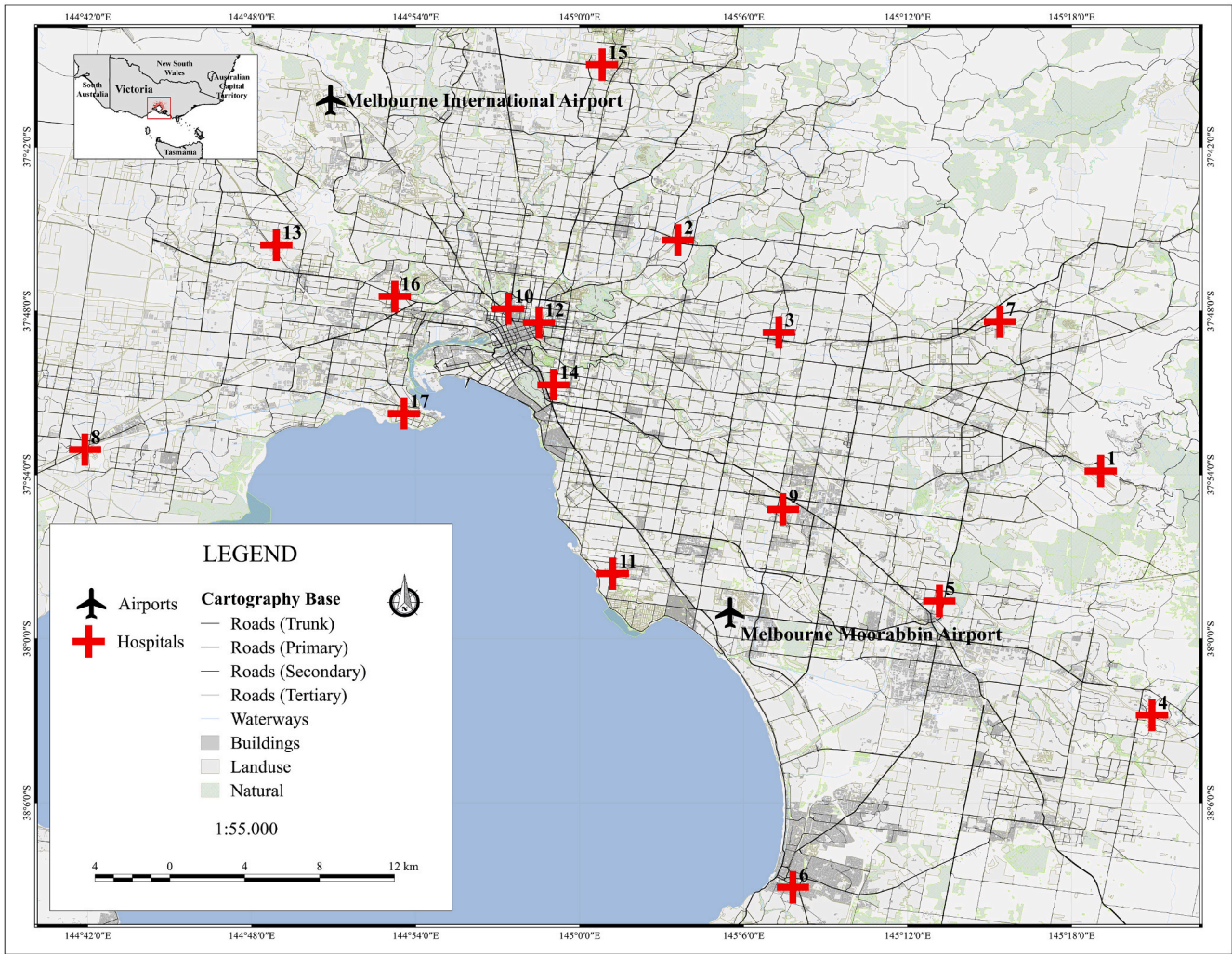


Fig. 4. Case study area- Greater Melbourne metropolitan area.

expendable resources considered for emergency operations were (a) medical supplies, and (b) surgical supplies. Also, the availability of these expendable resources at the Tullamarine and Moorabbin Airports were assumed as (6500, 300) and (3500, 200) units respectively. The assumption incorporated to the study concerning the demands (in units) for each type of resources at each hospital was based on the corresponding area’s population and the average number of emergencies and resuscitation patients per month that visited the emergency department at each hospital. The processing time (in hours) for the relief operations at each hospital have been considered according to the demand, e.g. 10 min per emergency patient as per the National Health Performance Authority. Huber-Wagner et al. [103] studied the duration’s necessary for emergency operations during mass casualty scenarios and revealed that a mean operation requires an average time of 130 min.

For the purpose of model validation, it was assumed that resuscitation patients require 130 min for treatment. Any extra volunteer or surgeon team member can reduce the processing time of an emergency operation to 10 and 15 min, respectively. The severity levels (importance levels of operations at each hospital) were generated randomly using discrete uniform distribution  $U(1, 6)$ . The details of the case study are tabulated in Table A3.

Table 7 shows the results of applying the proposed algorithms on the area of interest. The results of the analysis indicates that both Monte Carlo and GA algorithm could outperform other algorithms to calculate the total completion time of the emergency operations weighted by the

severity level. Computational time-wise, however, the results indicates that the Monte Carlo algorithm could outperform GA algorithm by 90% reduction in the CPU time. The Completion time of the case study is reported as 757 by Monte Carlo algorithm pertained in 187.92 s. The clustering algorithm could pertain objective value function of 838 within 7651 s which indicates that this algorithm is the less efficient method in the emergency response operations.

The detailed results of the case study is tabulated in Table 8. For instance,  $V_1$  picks up 25 health professionals and 20 surgeons with enough supply of the emergency expendable items from Melbourne airport to visit 5 hospitals including Western, Northern, St Vincent, Box hill, and Alfred hospitals, respectively. The generated visiting plan for the rest of points is shown in Table 8.

Table 8  
Case study resource allocation.

Vehicle	RC	Sequence	Expandable Items		Non-expandable Items	
			$T_1$	$T_2$	$R_1$	$R_2$
$V_1$	$RC_1$	$H_{16} \rightarrow H_{15} \rightarrow H_{12} \rightarrow H_3 \rightarrow H_{14}$	3045	176	25	20
$V_2$	$RC_1$	$H_9 \rightarrow H_6 \rightarrow H_{13} \rightarrow H_7$	2978	102	31	14
$V_3$	$RC_2$	$H_1 \rightarrow H_5 \rightarrow H_{17} \rightarrow H_4$	1644	70	17	9
$V_4$	$RC_2$	$H_{11} \rightarrow H_{10} \rightarrow H_2 \rightarrow H_8$	1692	100	20	10

## 6. Discussion and concluding remarks

At strategic level, during an emergency or disaster response, a relief centre is a central command and control facility that is responsible for preparing, planning, coordinating, and controlling the EROs. Strategic and operational decisions should be, however, made in a short time span to protect lives and properties. For this reason, it is crucial to employ effective tools and methods to facilitate swift and accurate planning for covering all incidents. In the majority of recovery operations, engaging non-expendable resources is inevitable. In bushfire emergency response, for example, both firefighters and medical crews should be dispatched to suppress the fire and assist injured people, simultaneously. A similar situation exists as part of the Flexible Home Aged Care package (hospital in home plan) where the operations team is required to attend a number of calls for assistance when several seniors require immediate attention.

In the vehicle routing problems that non-expendable resources are also involved in emergency response operations, the role of resource planning becomes utterly crucial. Based on collected data from incidents or direct communication with people who are urgently requiring help, the severity of incidents is assessed and the number of non-expendable resources are determined for each vehicle in a way that out of all allocated incidents, the one with the maximum demand is also covered. The implication of this arrangement is that for other incidents, extra non-expendable resources become available. It is obvious that extra available resources will not stay put and they will help provide assistance to process the incidents faster. This is where the main contribution of this study stands out: the estimated processing time of incidents should be updated in the planning process given that the objective function of scheduling models for this type of problem is generally formulated as the total weighted completion time of all vehicles. If this key factor is not taken into account, incompetent or poor planning will result in either over-supplying resources or delays in processing the incidents.

Adjusting processing time of recovery operations for each incident can be undertaken in different planning stages using various methods: (1) Solving the problem without adjusting the processing time (Greedy method) and then making the adjustments for all incidents and the total weighted completion times of all vehicles (Augmented Greedy method); (2) As a post-analysis technique and through an iterative method, the output of Augmented Greedy algorithm can be further improved (k-Node Crossover); (3) Incorporating the adjusted processing time in the decision-making process when selecting a vehicle to cover the next incident (Scheduling method); (4) Improving the decisions on incident selection in a post-analysis process via introducing a random process for choosing the best next incident to cover (Monte Carlo method); and (5) Breaking down the main problem into sub-problems (clusters) and using the exact method to solve the smaller sized problems. This is followed by aggregating the optimal values of objective function for all sub-problems, in order to obtain the optimal value for the main problem (Clustering algorithm).

Comparing the results of the Augmented Greedy method with the original Greedy algorithm clearly indicates that ignoring to update the processing time for non-expendable resources results in higher total weighted completion time. Therefore, this gap should be filled in planning and scheduling processes of recovery operations. The question here is whether updating the final schedule (Augmented Greedy method) and applying an improved methods such as k-Node Crossover is the most effective way to tackle this issue. Our results stipulate that although both k-Node Crossover methods demonstrate a superior performance compared to the Augmented Greedy algorithm, fine-tuning the decision-

making process of identifying the local optimal choice by considering the updated processing time of the next incident (i.e. Scheduling method) yields a significant and positive impact on the schedule. We have also noticed that since the Scheduling algorithm inherits the negative properties of the Greedy method with the possibility of losing a better optimal solution as a result of stage-based decision-making, a post-analysis method is more effective to address this shortcoming. That is the reason Monte Carlo has outperformed all heuristics and delivered the most effective solution. Furthermore, the comparison between the percentage of improvement among each pair of heuristics reveals that heuristics with post optimal improvements provide superior performance compared to the original models. Moreover, introducing the randomness into the decision model results in an improved solution, especially for scenarios that the number of allocated incidents to vehicles is higher.

In addition to the aforementioned findings, we benchmarked the solutions against the meta-heuristic (GA) and clustering algorithms which have been cited many times for planning of EROs. It is, however, crucial to note that the planning process should be undertaken within a reasonable time span to address the requirements of managing recovery operations. From the computational time perspective, all scenarios have been successfully analysed by the six developed heuristics in less than 30 min, which is in the acceptable range. The only exception is the largest scenarios which took almost 48 min to analyse by the Monte Carlo method. That explains why we limit the GA to provide the solution in 30 min. As indicated in section 5.3, GA was not successful in providing a better solution compared to the heuristics within the specified time limit. Therefore, although GA provides a reasonable solution for planning of various operations across supply chains, for a limited planning time span, some reservations should be taken into consideration.

It is also recommended by several studies to use the Clustering algorithm for planning the recovery operations. The logic behind this recommendation is that although breaking down the main problem into sub-problems introduces sub-optimal routing solutions, using the exact method for solving each cluster might make up the sub-optimality of solutions for each portion of problem. Although we have developed a novel heuristic method to effectively cluster the problems into more relevant sub-problems using the relatedness function between incidents, results show no significant advantage of the Clustering algorithm to heuristics. In addition, it is worth mentioning that solving the problem using the iterative Clustering algorithm could take up more than two days to deliver the final solution. As a result, we recommend avoiding Clustering algorithms for scheduling of recovery operations where the planning period should be sufficiently short.

Probing the computational time of the defined bundle of base and post-analysis improving methods reveals that the first group including the combination of Greedy, Augmented Greedy, and k-Node Crossover methods is less affected by the size of the problem. Basically, the first procedure is recommended for the situations that time is pressing and the second procedure would be more appropriate when a superior solution is preferred at the cost of spending a little bit more time on the planning process. Scheduling, however, could be considered as a conservative approach whereby a reasonable solution is delivered within a short time span.

To summarise the findings of this study into a guideline, managers of emergency recovery operations are recommended to (i) apply the recommended adjustment on operating time. Availability of extra non-expendable resources for less severe incidents makes these incidents to be covered faster. In other word, one of the proposed algorithms of this

study should be employed in the planning process; (ii) avoid using the clustering method to plan the recovery operations; (iii) employ the procedure 2 for planning when precision of solution is more important than the urgency of delivering the solution. If 30 min of computational time for planning process is within the acceptable range, managers of ERO are recommended to use the procedure 2 when the problem size is less than 1200; (iv) use procedure 1 for larger problem size or very short time frames of planning process. Following these recommendations provides significant benefits to managers of ERO such as optimal deployment of resources, obtaining more accurate processing time, decreasing the likelihood of over-supplying non-expendable resources and thus reducing the cost of operations, and effective selection of a solution approach based on the trade-off between the computational time and optimality of the solution.

To further improve the developed methods, future studies can extend

the proposed algorithms by incorporating the stochastic variables and conducting stochastic optimisation to address the uncertainties involved in scheduling of recovery operations. Another promising pathway is to develop multi-objective models in which the trade-off between the processing time and severity level of incidents for selecting the next incident to cover is investigated. Furthermore, employing the decomposition methods to solve the exact model of this problem for large sized emergency scenarios might be of great value.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Appendix A. Computational time of Heuristics**

Table A.1  
Computation time and 95% confidence interval of heuristics (in seconds)

Scenario	Parameter	Greedy	Updated Greedy	2-N Crossover	3-N Crossover	Scheduling	Monte Carlo
1	$\mu$	0.06	0.05	10.50	10.34	0.07	165.09
	95% CI - LB	0.05	0.04	10.45	10.24	0.07	164.16
	95% CI - UB	0.07	0.07	10.55	10.43	0.07	166.02
2	$\mu$	0.05	0.03	10.54	10.50	0.07	171.14
	95% CI - LB	0.05	0.03	10.49	10.43	0.07	170.30
	95% CI - UB	0.05	0.03	10.58	10.56	0.07	171.98
3	$\mu$	0.08	0.06	23.37	23.20	0.12	347.76
	95% CI - LB	0.08	0.06	23.27	23.09	0.12	345.66
	95% CI - UB	0.09	0.06	23.46	23.30	0.13	349.86
4	$\mu$	0.09	0.06	23.92	23.78	0.12	370.28
	95% CI - LB	0.09	0.06	23.85	23.69	0.12	368.13
	95% CI - UB	0.09	0.06	23.98	23.87	0.13	372.44
5	$\mu$	0.14	0.09	42.28	41.99	0.18	615.43
	95% CI - LB	0.13	0.08	42.19	41.84	0.18	609.81
	95% CI - UB	0.14	0.09	42.37	42.13	0.19	621.05
6	$\mu$	0.15	0.10	43.72	43.32	0.20	697.27
	95% CI - LB	0.15	0.09	43.61	43.17	0.20	693.75
	95% CI - UB	0.16	0.10	43.82	43.47	0.21	700.79
7	$\mu$	0.30	0.18	99.10	99.28	0.39	1572.03
	95% CI - LB	0.29	0.17	98.76	98.99	0.38	1561.42
	95% CI - UB	0.30	0.19	99.43	99.58	0.40	1582.63
8	$\mu$	0.32	0.18	102.08	101.91	0.40	1670.65
	95% CI - LB	0.31	0.18	101.79	101.68	0.40	1661.59
	95% CI - UB	0.33	0.19	102.37	102.14	0.41	1679.71
9	$\mu$	1.85	0.28	175.93	175.89	2.13	1847.72
	95% CI - LB	1.80	0.27	175.44	175.22	2.10	1827.60
	95% CI - UB	1.90	0.29	176.42	176.56	2.17	1867.84
10	$\mu$	2.99	0.36	207.51	207.92	3.22	2833.32
	95% CI - LB	2.91	0.33	206.68	207.01	3.17	2812.78
	95% CI - UB	3.07	0.39	208.35	208.82	3.27	2853.87

Table A.2  
Summary of 17 public hospitals considered in the case study simulation.

No.	Hospital name	No. of beds	Address	Location	Post Code
1	Angliss Hospital	100–199	Albert Street	Upper Ferntree Gully	3156
2	Austin Hospital	> 500	145 Studley Road	Heidelberg	3084
3	Box Hill Hospital	200–500	51 Nelson Road	Box Hill	3128
4	Casey Hospital	200–500	52 Kangan Drive	Berwick	3806
5	Dandenong Hospital	200–500	105-135 David Street	Dandenong	3175
6	Frankston Hospital	200–500	Hastings Rd	Frankston	3199
7	Maroondah Hospital	200–500	1 Mt Dandenong Rd	East Ringwood	3135
8	Mercy Hospital	100–199	300 Princes Highway	Werribee	3030
9	Monash M.C.	>500	246 Clayton Road	Clayton	3168
10	Royal Melbourne	>500	300 Grattan Street	Parkville	3050
11	Sandringham Hospital	100–150	193 Bluff Road	Sandringham	3191
12	St Vincent’s Hospital	>500	41 Victoria Parade	Fitzroy	3065

(continued on next page)

(continued)

No.	Hospital name	No. of beds	Address	Location	Post Code
13	Sunshine Hospital	200–500	176 Furlong Road	St. Albans	3021
14	Alfred Hospital	> 500	55 Commercial Road	Melbourne	3004
15	The Northern Hospital	200–500	185 Cooper Street	Epping	3076
16	Western Hospital	200–500	160 Gordon Street	Footscray	3011
17	Williamstown Hospital	50–99	77b Railway Crescent	Williamstown	3016

Table A.3  
Details of the case study.

H. No.	Expendable resources		Non-expendable resources		Processing times		Severity $W_i$
	Medical supplies	Surgical supplies	Health professionals	Surgeons	Non-surgery emergencies	Surgery resuscitation	
	$T = 1$	$T = 2$	$R = 1$	$R = 2$	$P_i^1$	$P_i^2$	
1	337	12	8	4	7.02	6.51	6
2	790	29	20	10	6.58	6.29	3
3	664	25	17	8	6.51	6.78	2
4	467	9	12	3	6.49	6.51	1
5	690	28	17	9	6.76	6.75	5
6	1238	33	31	11	6.66	6.51	6
7	464	19	12	6	6.44	6.87	2
8	176	3	4	1	7.33	6.51	1
9	920	41	23	14	6.67	6.36	6
10	572	62	14	21	6.81	6.41	4
11	154	4	4	1	6.42	8.68	6
12	311	25	8	8	6.48	6.78	3
13	356	9	9	3	6.59	6.51	2
14	657	60	16	20	6.84	6.51	1
15	997	40	25	13	6.65	6.68	5
16	416	26	10	9	6.93	6.27	6
17	150	21	4	7	6.25	6.51	2

Appendix B. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.ijdr.2020.101780>.

References

[1] L. Lei, M. Pinedo, L. Qi, S. Wang, J. Yang, Personnel scheduling and supplies provisioning in emergency relief operations, *Ann. Oper. Res.* 235 (2015) 487–515.

[2] B. Teague, R. McLeod, S.M. Pascoe, et al., 2009 Victorian Bushfires Royal Commission: Interim Report 2: Priorities for Building in Bushfire Prone Areas, Government Printer, 2009.

[3] K. Lee, L. Lei, M. Pinedo, S. Wang, Operations scheduling with multiple resources and transportation considerations, *Int. J. Prod. Res.* 51 (2013) 7071–7090.

[4] B. Bodaghi, E. Palaneeswaran, B. Abbasi, Bi-objective multi-resource scheduling problem for emergency relief operations, *Prod. Plann. Contr.* 29 (2018) 1191–1206.

[5] H. Arora, T. Raghu, A. Vinze, Resource allocation for demand surge mitigation during disaster response, *Decis. Support Syst.* 50 (2010) 304–315.

[6] E. Rolland, R.A. Patterson, K. Ward, B. Dodin, Decision support for disaster management, *Oper. Manag. Res.* 3 (2010) 68–79.

[7] H. Hasanzadeh, M. Bashiri, An efficient network for disaster management: model and solution, *Appl. Math. Model.* 40 (2016) 3688–3702.

[8] C. Hu, X. Liu, Y. Hua, A bi-objective robust model for emergency resource allocation under uncertainty, *Int. J. Prod. Res.* 54 (2016) 7421–7438.

[9] N. Sahebjamnia, S.A. Torabi, S.A. Mansouri, A hybrid decision support system for managing humanitarian relief chains, *Decis. Support Syst.* 95 (2017) 12–26.

[10] M. Traub, D.A. Bradt, A.P. Joseph, The surge capacity for people in emergencies (scope) study in australasian hospitals, *Med. J. Aust.* 186 (2007) 394–398.

[11] H. Sheikhbardsiri, A.R. Raeisi, M. Nekoei-Moghadam, F. Rezaei, Surge capacity of hospitals in emergencies and disasters with a preparedness approach: a systematic review, *Disaster Med. Public Health Prep.* 11 (2017) 612–620.

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

[13] G. Laporte, A concise guide to the traveling salesman problem, *J. Oper. Res. Soc.* 61 (2010) 35–40.

[14] A.M. Campbell, D. Vandenbussche, W. Hermann, Routing for relief efforts, *Transport. Sci.* 42 (2008) 127–145.

[15] F. Wex, G. Schryen, S. Feuerriegel, D. Neumann, Emergency response in natural disaster management: allocation and scheduling of rescue units, *Eur. J. Oper. Res.* 235 (2014) 697–708.

[16] O. Rodríguez-Espíndola, P. Albores, C. Brewster, Disaster preparedness in humanitarian logistics: a collaborative approach for resource management in floods, *Eur. J. Oper. Res.* 264 (2018) 978–993.

[17] V. Pillac, P. Van Hentenryck, C. Even, A conflict-based path-generation heuristic for evacuation planning, *Transp. Res. Part B Methodol.* 83 (2016) 136–150.

[18] S. Shahparvari, B. Abbasi, P. Chhetri, Possibilistic scheduling routing for short-notice bushfire emergency evacuation under uncertainties: an Australian case study, *Omega* 72 (2017) 96–117.

[19] M. Bruni, P. Beraldi, S. Khodaparasti, A fast heuristic for routing in post-disaster humanitarian relief logistics, *Transport. Res. Procedia* 30 (2018) 304–313.

[20] S. Zhang, X. Ruan, Y. Xia, X. Feng, Foldable container in empty container repositioning in intermodal transportation network of belt and road initiative: strengths and limitations, *Marit. Pol. Manag.* 45 (2018) 351–369.

[21] K. Schneider, S.G. Nurre, A multi-criteria vehicle routing approach to improve the compliance audit schedule for food banks, *Omega* 84 (2019) 127–140.

[22] L. Bertazzi, L.C. Coelho, A. De Maio, D. Laganà, A matheuristic algorithm for the multi-depot inventory routing problem, *Transport. Res. E Logist. Transport. Rev.* 122 (2019) 524–544.

[23] D. Li, L. Ding, S. Connor, When to switch? index policies for resource scheduling in emergency response, *Prod. Oper. Manag.* 29 (2020) 241–262.

[24] K.S. Jung, M. Pinedo, C. Sriskandarajah, V. Tiwari, Scheduling elective surgeries with emergency patients at shared operating rooms, *Prod. Oper. Manag.* 28 (2019) 1407–1430.

[25] R. Mahmoudi, S.N. Shetab-Boushehri, S.R. Hejazi, A. Emrouznejad, P. Rajabi, A hybrid egalitarian bargaining game-dea and sustainable network design approach for evaluating, selecting and scheduling urban road construction projects, *Transport. Res. E Logist. Transport. Rev.* 130 (2019) 161–183.

[26] Y. Shi, T. Boudouh, O. Grunder, A robust optimization for a home health care routing and scheduling problem with consideration of uncertain travel and service times, *Transport. Res. E Logist. Transport. Rev.* 128 (2019) 52–95.

[27] M. Bababeik, N. Khademi, A. Chen, Increasing the resilience level of a vulnerable rail network: the strategy of location and allocation of emergency relief trains, *Transport. Res. E Logist. Transport. Rev.* 119 (2018) 110–128.

[28] X. Wu, Y. Cao, Y. Xiao, J. Guo, Finding of urban rainstorm and waterlogging disasters based on microblogging data and the location-routing problem model of urban emergency logistics, *Ann. Oper. Res.* (2018) 1–32.

[29] L. Zhu, Y. Gong, Y. Xu, J. Gu, Emergency relief routing models for injured victims considering equity and priority, *Ann. Oper. Res.* (2018) 1–34.

[30] A.M. Caunhye, X. Nie, S. Pokharel, Optimization models in emergency logistics: a literature review, *Soc. Econ. Plann. Sci.* 46 (2012) 4–13.



- [31] G. Galindo, R. Batta, Review of recent developments in or/ms research in disaster operations management, *Eur. J. Oper. Res.* 230 (2013) 201–211.
- [32] L. Özdamar, M.A. Ertem, Models, solutions and enabling technologies in humanitarian logistics, *Eur. J. Oper. Res.* 244 (2015) 55–65.
- [33] L. Zhou, X. Wu, Z. Xu, H. Fujita, Emergency decision making for natural disasters: an overview, *Int. J. Disaster Risk Reduct.* 27 (2018) 567–576.
- [34] A. Haghani, S.C. Oh, Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations, *Transport. Res. Pol. Pract.* 30 (1996) 231–250.
- [35] L. Özdamar, E. Ekinci, B. Küçükyaşar, Emergency logistics planning in natural disasters, *Ann. Oper. Res.* 129 (2004) 217–245.
- [36] M.S. Chang, Y.L. Tseng, J.W. Chen, A scenario planning approach for the flood emergency logistics preparation problem under uncertainty, *Transport. Res. E Logist. Transport. Rev.* 43 (2007) 737–754.
- [37] B. Balçık, B.M. Beamon, K. Smilowitz, Last mile distribution in humanitarian relief, *J. Intell. Transport. Syst.* 12 (2008) 51–63.
- [38] K. Huang, Y. Jiang, Y. Yuan, L. Zhao, Modeling multiple humanitarian objectives in emergency response to large-scale disasters, *Transport. Res. E Logist. Transport. Rev.* 75 (2015) 1–17.
- [39] S. Zokaei, A. Bozorgi-Amiri, S.J. Sadjadi, A robust optimization model for humanitarian relief chain design under uncertainty, *Appl. Math. Model.* 40 (2016) 7996–8016.
- [40] M. Falasca, C. Zobel, An optimization model for volunteer assignments in humanitarian organizations, *Soc. Econ. Plann. Sci.* 46 (2012) 250–260.
- [41] K. Lassiter, A. Khademi, K.M. Taafe, A robust optimization approach to volunteer management in humanitarian crises, *Int. J. Prod. Econ.* 163 (2015) 97–111.
- [42] G. Schryen, G. Raucher, T. Comes, Resource planning in disaster response, *Bus. Inf. Syst. Eng.* 57 (2015) 243–259.
- [43] B. Bodaghi, E. Palaneeswaran, An optimization model for scheduling emergency operations with multiple teams, in: *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Detroit, Michigan, 2016, pp. 436–442.
- [44] S. Shahparvari, B. Bodaghi, Risk reduction for distribution of the perishable rescue items; a possibilistic programming approach, *Int. J. Disaster Risk Reduct.* 31 (2018) 886–901.
- [45] M. Huang, K. Smilowitz, B. Balçık, Models for relief routing: equity, efficiency and efficacy, *Transport. Res. E Logist. Transport. Rev.* 48 (2012) 2–18.
- [47] A.M. Caunhye, Y. Zhang, M. Li, X. Nie, A location-routing model for prepositioning and distributing emergency supplies, *Transport. Res. E Logist. Transport. Rev.* 90 (2016) 161–176.
- [48] B. Bodaghi, E. Palaneeswaran, S. Shahparvari, M. Mohammadi, Probabilistic allocation and scheduling of multiple resources for emergency operations; a victorian bushfire case study, *Comput. Environ. Urban Syst.* 81 (2020) 101479.
- [49] P.P. Repoussis, C.D. Tarantilis, G. Ioannou, The open vehicle routing problem with time windows, *J. Oper. Res. Soc.* 58 (2007) 355–367.
- [50] S. MirHassani, N. Abolghasemi, A particle swarm optimization algorithm for open vehicle routing problem, *Expert Syst. Appl.* 38 (2011) 11547–11551.
- [51] S. Yu, C. Ding, K. Zhu, A hybrid ga-ts algorithm for open vehicle routing optimization of coal mines material, *Expert Syst. Appl.* 38 (2011) 10568–10573.
- [52] K. Fleszar, I.H. Osman, K.S. Hindi, A variable neighbourhood search algorithm for the open vehicle routing problem, *Eur. J. Oper. Res.* 195 (2009) 803–809.
- [53] R. Baños, J. Ortega, C. Gil, A. Fernández, F. De Toro, A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows, *Expert Syst. Appl.* 40 (2013) 1696–1707.
- [54] J. Tang, Y. Ma, J. Guan, C. Yan, A max–min ant system for the split delivery weighted vehicle routing problem, *Expert Syst. Appl.* 40 (2013) 7468–7477.
- [55] C. Tarantilis, C. Kiranoudis, Distribution of fresh meat, *J. Food Eng.* 51 (2002) 85–91.
- [56] R. Liu, Z. Jiang, N. Geng, A hybrid genetic algorithm for the multi-depot open vehicle routing problem, *OR spectrum* 36 (2014) 401–421.
- [57] K. Pichka, B. Ashjari, A. Ziaefar, P. Nickbeen, Open vehicle routing problem optimization under realistic assumptions, *Int. J. Res. Ind. Eng.* 3 (2014) 46–55.
- [58] P.A.M. Duque, I.S. Dolinskaya, K. Sørensen, Network repair crew scheduling and routing for emergency relief distribution problem, *Eur. J. Oper. Res.* 248 (2016) 272–285.
- [59] B. Fontem, S.H. Melouk, B.B. Keskin, N. Bajwa, A decomposition-based heuristic for stochastic emergency routing problems, *Expert Syst. Appl.* 59 (2016) 47–59.
- [60] M.S. Osman, B. Ram, Routing and scheduling on evacuation path networks using centralized hybrid approach, *Comput. Oper. Res.* 88 (2017) 332–339.
- [61] T.I. Faiz, C. Vogiatzis, M. Noor-E-Allah, A column generation algorithm for vehicle scheduling and routing problems, *Comput. Ind. Eng.* 130 (2019) 222–236.
- [62] A. Moreno, P. Munari, D. Alem, A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration, *Eur. J. Oper. Res.* 275 (2019) 16–34.
- [63] V. Bélanger, A. Ruiz, P. Soriano, Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles, *Eur. J. Oper. Res.* 272 (2019) 1–23.
- [64] V.A. Knight, P.R. Harper, L. Smith, Ambulance allocation for maximal survival with heterogeneous outcome measures, *Omega* 40 (2012) 918–926.
- [65] D. Liu, J. Han, J. Zhu, Vehicle routing for medical supplies in large-scale emergencies, *Lecture Notes Oper. Res.* 8 (2007) 412–419.
- [66] A. Ceselli, G. Righini, E. Trosoldi, Combined location and routing problems for drug distribution, *Discrete Appl. Math.* 165 (2014) 130–145.
- [67] F. Majzoubi, A Mathematical Programming Approach for Dispatching and Relocating EMS Vehicles, Ph.D. thesis, University of Louisville, 2014.
- [68] P. Tang, S. Zhu, Emergency logistics distribution vrp based on differential evolution optimization, *J. Software Eng.* 9 (2015) 105–113.
- [69] J. Zhao, Y. Guo, X. Duan, Dynamic path planning of emergency vehicles based on travel time prediction, *J. Adv. Transport.* 2017 (2017).
- [70] C. Ciancio, D. Laganà, R. Musmanno, F. Santoro, An integrated algorithm for shift scheduling problems for local public transport companies, *Omega* 75 (2018) 139–153.
- [71] T. Li, X. Wang, Study on the vehicle routing problem based on improved ga-greedy algorithm, *Logistics: the Emerging Frontiers of Transportation and Development in China*, 2009, pp. 2814–2822.
- [72] A. Almutairi, Sim-heuristic Algorithms for Robust Vehicle Routing Problem with Stochastic Demand, Ph.D. thesis, University of Portsmouth, 2016.
- [73] A. Bettinelli, A. Santini, D. Vigo, A real-time conflict solution algorithm for the train rescheduling problem, *Transp. Res. Part B Methodol.* 106 (2017) 237–265.
- [74] M. Kritikos, G. Ioannou, A greedy heuristic for the capacitated minimum spanning tree problem, *J. Oper. Res. Soc.* 68 (2017) 1223–1235.
- [75] F. Baptista, J. Tavares, Bio-inspired algorithms for the vehicle routing problem, in: *v INEGI, Panorama sociodemográfico de Morelos. México*, Springer, 2009, pp. 35–40.
- [76] Z. Zhang, M. Liu, A. Lim, A memetic algorithm for the patient transportation problem, *Omega* 54 (2015) 60–71.
- [77] Y.J. Zheng, S.Y. Chen, H.F. Ling, Evolutionary optimization for disaster relief operations: a survey, *Appl. Soft Comput.* 27 (2015) 553–566.
- [78] J.J. Jaw, A.R. Odoni, H.N. Psarafitis, N.H. Wilson, A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, *Transp. Res. Part B Methodol.* 20 (1986) 243–257.
- [79] M.X. Weng, J. Lu, H. Ren, Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective, *Int. J. Prod. Econ.* 70 (2001) 215–226.
- [80] S.D. Ramchurn, M. Polukarov, A. Farinelli, C. Truong, N.R. Jennings, Coalition formation with spatial and temporal constraints, in: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 3, International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1181–1188.
- [81] F. Wex, G. Schryen, D. Neumann, A fuzzy decision support model for natural disaster response under informational uncertainty, *Int. J. Inf. Syst. Crisis Response Manag.* 4 (2012) 23–41.
- [82] S. Abdullah, N.R. Sabar, M.Z.A. Nazri, M. Ayob, An exponential monte-carlo algorithm for feature selection problems, *Comput. Ind. Eng.* 67 (2014) 160–167.
- [83] H. Al-Harthei, V.T. Toth, A. Garib, S.A. Mahmoud, Efficient real-time allocation of patrol cars in traffic management, in: *Proceedings of the 2nd World Congress on Civil, Structural, and Environmental Engineering (CSSEE-17)*, 2017, pp. 1–8. ICTE–105.
- [84] F. Wu, R. Sioshansi, A two-stage stochastic optimization model for scheduling electric vehicle charging loads to relieve distribution-system constraints, *Transp. Res. Part B Methodol.* 102 (2017) 55–82.
- [85] B.M. Baker, M. Ayechev, A genetic algorithm for the vehicle routing problem, *Comput. Oper. Res.* 30 (2003) 787–800.
- [86] I. Okhrin, K. Richter, Vehicle routing problem with real-time travel times, *Int. J. Veh. Inf. Commun. Syst.* 2 (2009) 59–77.
- [87] K. Zidi, F. Mguis, P. Borne, K. Ghedira, Distributed genetic algorithm for disaster relief planning, *Int. J. Comput. Commun. Contr.* 8 (2013) 769–783.
- [88] F. Mguis, K. Zidi, K. Ghedira, P. Borne, Distributed approach for vehicle routing problem in disaster case, *IFAC Proc. Vol.* 45 (2012) 353–359.
- [89] J. Qin, Y. Ye, B.r. Cheng, X. Zhao, L. Ni, The emergency vehicle routing problem with uncertain demand under sustainability environments, *Sustainability* 9 (2017) 288.
- [90] L. Özdamar, O. Demir, A hierarchical clustering and routing procedure for large scale disaster relief logistics planning, *Transport. Res. E Logist. Transport. Rev.* 48 (2012) 591–602.
- [91] Y. He, J. Wen, M. Huang, Study on emergency relief vrp based on clustering and pso, in: *2015 11th International Conference on Computational Intelligence and Security (CIS)*, IEEE, 2015, pp. 43–47.
- [92] J. Vargas-Florez, R. Medina-Rodríguez, R. Alva-Cabrera, A clustering optimization approach for disaster relief delivery: a case study in lima-Peru, in: *Information Management and Big Data*, Springer, 2015, pp. 69–80.
- [93] Z. Gharib, A. Bozorgi-Amiri, R. Tavakkoli-Moghaddam, E. Najafi, A cluster-based emergency vehicle routing problem in disaster with reliability, *Sci. Iran. Trans. E, Ind. Eng.* 25 (2018) 2312–2330.
- [94] P.H.V. Penna, A.C. Santos, C. Prins, Vehicle routing problems for last mile distribution after major disaster, *J. Oper. Res. Soc.* 69 (2018) 1254–1268.
- [95] F. Arnold, K. Sørensen, What makes a vrp solution good? the generation of problem-specific knowledge for heuristics, *Comput. Oper. Res.* (2018).
- [96] U. Bellur, R. Kulkarni, Improved matchmaking algorithm for semantic web services based on bipartite graph matching, in: *IEEE International Conference on Web Services (ICWS 2007)*, IEEE, 2007, pp. 86–93.
- [97] W. Jiang, Analysis and discussion of radar construction problems with greedy algorithm, in: *Advances in Computer Communication and Computational Sciences*, Springer, 2019, pp. 303–312.
- [98] S. Shahparvari, B. Abbasi, P. Chhetri, A. Abareshi, Fleet Routing and Scheduling in Bushfire Emergency Evacuation: A Regional Case Study of the Black Saturday Bushfires in Australia, *Transportation Research Part D: Transport and Environment*, 2017.
- [99] S. Mardle, S. Pascoe, et al., An overview of genetic algorithms for the solution of optimisation problems, *Comput. High. Educ. Econ. Rev.* 13 (1999) 16–20.



- [100] G. Barbarosoğlu, L. Özdamar, A. Çetevik, An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations, *Eur. J. Oper. Res.* 140 (2002) 118–133.
- [101] J. Perez-Ortega, N.N. Almanza-Ortega, D. Romero, Balancing effort and benefit of k-means clustering algorithms in big data realms, *PLoS One* 13 (2018) 1–19.
- [102] B. Bodaghi, *Optimisation Approaches for Scheduling Disaster Emergency Operations with Multiple Resources*, Ph.D. thesis, SWINBURNE UNIVERSITY OF TECHNOLOGY Melbourne, Australia, 2017, 2017.
- [103] S. Huber-Wagner, R. Lefering, L.M. Qvick, M. Körner, M.V. Kay, K.J. Pfeifer, M. Reiser, W. Mutschler, K.G. Kanz, et al., on Polytrauma of the German Trauma Society, W.G., Effect of whole-body ct during trauma resuscitation on survival: a retrospective, multicentre study, *Lancet* 373 (2009) 1455–1461.