The isolated community evacuation problem with mixed integer programming

Klaas Fiete Krutein [a],[*], Anne Goodchild [b]

[a] *Department of Industrial & Systems Engineering, University of Washington, Box 352650, Seattle, WA 98195, United States of America*
[b] *Department of Civil & Environmental Engineering, University of Washington, 201 More Hall, Box 352700, Seattle, WA 98195, United States of America*

A R T I C L E   I N F O

A B S T R A C T

As awareness of the vulnerability of isolated regions to natural disasters grows, the demand for efficient evacuation plans is increasing. However, isolated areas, such as islands, often have characteristics that make conventional methods, such as evacuation by private vehicle, impractical to infeasible. Mathematical models are conventional tools for evacuation planning. Most previous models have focused on densely populated areas, and are inapplicable to isolated communities that are dependent on marine vessels or aircraft to evacuate. This paper introduces the Isolated Community Evacuation Problem (ICEP) and a corresponding mixed integer programming formulation that aims to minimize the evacuation time of an isolated community through optimally routing a coordinated fleet of heterogeneous recovery resources. ICEP differs from previous models on resource-based evacuation in that it is highly asymmetric and incorporates compatibility issues between resources and access points. The formulation is expanded to a two-stage stochastic problem that allows scenario-based optimal resource planning while also ensuring minimal evacuation time. In addition, objective functions with a varying degree of risk are provided, and the sensitivity of the model to different objective functions and problem sizes is presented through numerical experiments. To increase efficiency, structure-based heuristics to solve the deterministic and stochastic problems are introduced and evaluated through computational experiments. The results give researchers and emergency planners in remote areas a tool to build optimal evacuation plans given the heterogeneous resource fleets available, which is something they have not been previously able to do and to take actions to improve the resilience of their communities accordingly.

1. Introduction

1.1. Motivation

The new model formulation presented in this paper was motivated by the rising need to prepare for and mitigate the effects of disasters caused by natural hazards on the populations in remote communities. Particularly, small inhabited islands and similarly isolated communities such as coastal communities, remote valley hamlets, and mountain towns are vulnerable to the effects of natural disasters because of their dependence on waterways or limited and vulnerable roads. These conditions often do not allow the affected population to evacuate in private vehicles. Therefore, emergency management authorities often need to coordinate a

---

highly heterogeneous set of recovery resources that have to alternate between the disaster area and shelter locations to evacuate the entire population. Some disasters that require such actions are wildfires, such as the Australian bushfires, which required a marine and air evacuation of Mallacoota in Victoria, Australia in 2020 (Australian Broadcasting Corporation, 2020), and the evacuation of Samos Island in Greece during the wildfires of 2019 (Coffey, 2019). Storms can also require such an evacuation, such as in the Bahamas after Hurricane Dorian in 2019 (Romero, 2019). Volcanic eruptions led to the evacuation of Vulcano Island in Italy in 2021 (Nadeau, 2021), the evacuation of St. Vincent in 2021 (Deane and Coto, 2021), and the evacuation of some islands in Tonga in 2022 (Moussa and Rising, 2022). These events increased awareness of threats and caused remote communities to recognize the need to develop robust and quick evacuation plans (Britten, 2019) at a time when climate change is increasing the risk of many natural disasters (IPCC, 2012). Furthermore, communities with geographically vulnerable characteristics are not uncommon. In fact, a geospatial data analysis conducted by StreetLight Data (StreetLight Data Inc., 2020) has highlighted the 100 most difficult to evacuate communities in the United States, most of which are either in remote areas in the mountains or in coastal settings such as islands or peninsulas, where road-based evacuation is not possible. No existing formulation is capable of capturing this exact problem, and thus, a new formulation is required. This problem can be expressed through two research questions:

1. *During an emergency, how can resources be optimally routed to evacuate the entire community as quickly as possible?*
2. *During evacuation planning, which resources need to be secured to prepare for quick evacuation over a variety of disaster scenarios?*

For both questions, the compatibility between recovery resources and landing locations needs to be considered. For the second question, any decision has to be made without exact information about the nature of the disaster or evacuation demand patterns. Since this problem is complex when multiple pick up locations are considered, mathematical modeling is the right solution approach.

This paper refers to the problem that poses the two research question above as the Isolated Community Evacuation Problem (ICEP). The paper provides two formulations: a deterministic mixed-integer programming formulation (D-ICEP), and a two-stage stochastic mixed integer formulation with recourse (S-ICEP). The D-ICEP can be used for optimizing the evacuation plan for an isolated community, where all parameter and set data is known in advance and corresponds to the first research question. The recovery resources under consideration can be, depending on the environment to which the model is applied, a heterogeneous fleet of marine vessels, aircraft or land vehicles. The D-ICEP can therefore be used for response purposes and help decision makers and emergency managers to make decisions on how to effectively allocate available recovery resources to different parts of the disaster area and how to evacuate the affected population in the fastest possible way. It will also give insights into which part of the area will be most difficult to evacuate and where more resources can potentially help reduce the evacuation time further.

The S-ICEP is an expansion of the D-ICEP for planning purposes, which adds a resource selection decision that is relevant to answer the second research question. A two-stage stochastic programming formulation with recourse makes it possible to separate the problem into two stages, separated by a probabilistic event: the disaster causing the evacuation. The S-ICEP therefore provides emergency planning teams with a way to plan for evacuations and to evaluate the community's level of preparedness for evacuation scenarios of different natures. Furthermore, decision makers can choose between multiple objective functions for the S-ICEP that balance the conflicting objectives of time and cost in different ways. An analysis with the S-ICEP can help communities decide whether the current infrastructure is sufficient to support a timely evacuation. For example, it will help to identify which areas are most vulnerable to disaster, and whether it could be helpful to reactivate a decommissioned air strip, upgrade docking infrastructure for vessels, or whether additional recovery resources need to be held available to be prepared for a disaster. It could also help identify which gathering points people should travel to, to ensure evacuation can be executed as quickly as possible. Investigating all potential managerial insights and the resulting requirements for data inputs and disaster scenario design in practice would go beyond the scope of this paper. Krutein et al. (2022) provide a first deep dive into how to effectively use evacuation models to gain managerial insights through a real-world case study. In addition to introducing the model formulations, bounds are established on some key parameters that illustrate the model dynamics of both formulations, and some numerical experiments are provided. For an alternative efficient solution process, structure-based heuristics are presented for both the D-ICEP and the S-ICEP, and are evaluated numerically.

1.2. Related work

1.2.1. General framework

To assess where the ICEP falls in a general evacuation framework, Tüydeş (2005) provided multiple components of a general evacuation study. The first components are a hazard analysis, which investigates the severity of the event, and a vulnerability analysis, which identifies the population at risk. On the basis of the outcome of these components, disaster response actions can be defined (Tüydeş, 2005). These include emergency operations and evacuation coordination, which both involve the aspect of traffic management and coordination. Southworth (1991) further mentioned the need to conduct a behavior analysis of the population during an evacuation and a shelter analysis to identify where and how many shelters would be needed (Southworth, 1991). The ICEP falls into the response action component of evacuation analysis, which includes the planning and coordination of evacuation resources. For the ICEP, it can be assumed that sufficient information has been collected about the vulnerability of the population and that realistic disaster scenarios can be mapped. While consideration of evacuation behavior is important if the population can self-evacuate (Thompson et al., 2017), which describes the process where an evacuee can leave a dangerous area by either walking or in their personal vehicle, the ICEP does not provide the majority of the population with the option to leave the area entirely on their own.

In considering modeling approaches for the response action component of evacuation studies, Bayram's literature review (Bayram, 2016) provided a comprehensive field survey about the optimization of emergency planning. Evacuation models can be classified on the basis of their modeling approach, whether they are static or dynamic, whether they contain multiple levels, whether they are of stochastic or deterministic nature, whether shelter location decisions are included, and which modes of transportation are used.

1.2.2. Network flow problems

Network problems that aim to minimize the total route completion time, also known as the network clearance time, are particularly relevant. The simplest network flow model that describes this problem is the quickest flow model (Burkard et al., 1993). Building on this for evacuation applications, Lu et al. (2005) have used the time-expanded network flow model as a baseline for a new heuristic to achieve suboptimal solutions for the network clearance time for evacuations. To provide some recent examples, Lim et al. (2015) considered network clearance time in the context of reliability-based evacuation routing that included uncertainty in the link capacities caused by congestion. Pillac et al. (2016) developed a column generation-based two-level evacuating algorithm in which the sub-problem generates an evacuation path for each evacuation area, while the master problem resolves conflicts between the paths. Karabuk and Manzour (2019) have developed a multi-stage stochastic program for tornado evacuation management that considers the path uncertainty of a tornado to make evacuation decisions as the weather event evolves.

1.2.3. Related vehicle routing problems

It is worth exploring related problem types. The general vehicle routing problem (VRP) (Dantzig and Ramser, 1959), and its dynamic expansion (Laporte et al., 1992) (both generalizations of the Traveling Salesman Problem (TSP) Flood, 1956), are well-studied problems with a vast literature (Laporte, 2009; Pillac et al., 2013). Similarities to the ICEP can be found with the VRP with time windows (VRP-TW) (Schrage, 1981), a generalization of the VRP. Another related problem class is the location-routing problem (LRP), which considers both the optimization of the vehicle routes and depot locations, see for example (Belenguer et al., 2011). This is particularly the case if shelter location considerations are part of the problem. Another related problem class is that of the multi-trip vehicle routing problem (MVRP), where vehicles can perform multiple round-trips and visit nodes multiple times to fulfill their orders (Brandão and Mercer, 1998; Cattaruzza et al., 2016). As a variant of the MVRP, the VRP with satellite depots can also be considered. In this problem, vehicles can take on additional orders at satellite facilities and do not have to go all the way back to the depot to take on additional orders (Crevier et al., 2007). Another related problem that is derived from the VRP-TW is the Dial-a-Ride-Problem (DARP) (Cordeau, 2006), where the vehicles share their capacity between multiple customers to transport customers from requested pick-up to requested drop-off points. However, this problem is strongly constrained by the time windows and maximum duration constraints for customers and therefore difficult to solve efficiently.

1.2.4. Related evacuation transit routing problems

No previous research has developed optimization models for non-road-based evacuations. Hence, structurally related network models for the evacuation of populations through the use of buses, trains, or other public transit vehicles (Renne et al., 2011) are reviewed. Mass transportation models optimize the routing of vehicles to evacuate an otherwise immobile population through a set of nodes representing evacuation area pick-up points and shelters. These models require routing decisions for the recovery resources, which makes them difficult to solve, but they enable exact modeling of the resource usage. However, only a few applicable papers on mass transit evacuation have been published.

Song et al. (2009) presented a location-routing problem (LRP) that models evacuations from an urban city network via transit vehicles and uses different heuristic and algorithmic approaches to solve the problem. Sayyady and Eksioglu (2010) provided a mixed-integer linear program (MILP) that optimizes the total evacuation time in urban areas for no-notice evacuation by using buses that collect passengers from multiple pick-up locations until the bus capacity has been reached. An et al. (2013) expanded the bus evacuation idea and integrated it with decisions on the design of evacuation pick-up locations and furthermore considered service availability. Abdelgawad and Abdulhai (2010) developed a large-scale multi-modal evacuation model that combines a private vehicle evacuation model with a VRP-based mass transit evacuation model to obtain a holistic evacuation plan for large cities (NETDC). Kulshrestha et al. (2014) expanded this problem by considering pick-up location decisions and incorporating demand uncertainty through a robust optimization formulation. An additional study by Wu et al. (2020) considers evacuation by barges (BEPP) ahead of storm events.

Bish (2011) presented a bus-based evacuation model as a variant of the general VRP, called the bus evacuation problem (BEP). The BEP requires a significantly different formulation than the classic VRP, but it uses a network structure that is similar to that of the ICEP. It consists of initial bus depots, pick-up points, and shelter locations. Buses are routed from an initial depot to the pick-up points and then alternate between the pick-up points and shelter locations, sequentially evacuating the entire population. Multiple extensions to the BEP have been published. Pereira and Bish (2015) expanded the BEP to consider the arrival rates of people at pick-up points. Zheng (2014) provided a similar model to optimize mass transit evacuations of urban areas with different constraints on arrivals of evacuees, and solved it by using a Lagrangian relaxation-based algorithm. Goerigk et al. (2013) introduced multiple solution approaches that use branch-and-bound procedures to solve the BEP efficiently, and presented a simplified robust optimization formulation of this problem (RBEP) with delayed scenario generation (Goerigk and Grün, 2014). This formulation solves the RBEP for uncertain numbers of evacuees by using a linear and a tabu search to find near-optimal solutions. The search procedures take advantage of the problem's high symmetry, caused by identical buses and symmetric travel times. Goerigk et al. (2015) further developed a way to solve the RBEP by using two stages that sequentially add new scenarios generated from uncertainty set, solving

Table 1
Feature evaluation of selected related models and studies.

Study	Model	Multiple resources	Routing element	Flow element	Different node classes	Individual routes per resource	Limited compatibility	Multiple trips per resource	Different routes per trip	Heterogeneous fleets	Partial self-evacuation	Asymmetric travel times	Minimizing total duration of route plan	Demand uncertainty
Flood (1956)	TSP		✓			(✓)						(✓)		
Dantzig and Ramser (1959)	VRP(-TW)	✓	✓			✓						(✓)		
Belenguer et al. (2011), Song et al. (2009)	LRP	✓	✓		✓	✓								
Cattaruzza et al. (2016)	MVRP	✓	✓			✓		✓	✓					
Cordeau (2006)	DARP	✓	✓		✓									
Sayyady and Eksioglu (2010)	NETDC	✓		✓	✓			✓						
Wu et al. (2020)	BEPP	✓	✓		✓	✓		✓	✓	✓			✓	
Bish (2011)	BEP	✓	✓	✓	✓	✓		✓	✓				✓	
Goerigk and Grün (2014)	RBEP	✓	✓	✓	✓	✓		✓	✓				✓	✓
Dikas and Minis (2016)	CEP	✓	✓	✓	✓	✓		✓	✓					

a much larger number of uncertain scenarios in a reasonable amount of time. Goerigk et al. (2014b) further improved the optimal evacuation time of the network by combining the BEP with decisions on pick-up locations. Lastly, Goerigk et al. (2014a) integrated the BEP into a comprehensive evacuation framework that not only considers the aspects of previous work (Goerigk et al., 2014b), but introduces multi-modal commodity decisions and solves the problem by using a genetic algorithm.

Dikas and Minis (2016) expanded the usage of the BEP to the recovery of casualties, such as in a ceasefire on a battlefield, and created a variant called the Casualty Evacuation Problem (CEP). They further provided a hybrid solution framework for the BEP that combines the BEP heuristic concepts (Bish, 2011) with column-generation (Dikas and Minis, 2016). Baou et al. (2018) introduced a variant of the BEP that can consider heterogeneous bus capacity and take into account mobility impairments among some evacuees. Lastly, Wang and Wang (2019) consider re-balancing both supply and demand across the evacuation locations in the BEP to reduce total evacuation time.

1.2.5. Gaps in literature

No previous research has provided any solutions for optimizing the evacuation of isolated communities. Table 1 visualizes a comparison of selected related studies and structurally similar models evaluating them against the features that characterize the ICEP. Common routing problem formulations that are derived from the TSP and VRP do not contain the constraints required to model the network structure, the resource heterogeneity, and the objective functions that are needed for the ICEP. Furthermore, while structurally similar problems, particularly the BEP (Bish, 2011) and its variants (Goerigk et al., 2014b; Goerigk and Grün, 2014; Dikas and Minis, 2016; Baou et al., 2018), are useful as a baseline for formulating and solving the ICEP, no other published research has focused on the specific circumstances of geographically isolated areas. This includes the partial incompatibility between resources and access points, multiple access point alternatives per location, asymmetric travel time matrices, heterogeneous speed and capacity capabilities of recovery resources, and, for the stochastic cases, the challenges to predict how a disaster will unfold. The formulations presented in this paper aim to fill these gaps.

1.3. Contributions

Despite the increased demand for evacuation plans for vulnerable isolated communities, the reviewed literature on optimal evacuation modeling has focused primarily on urban evacuations on road networks. Bayram (2016) further found that most researchers have considered the management of emergency response resources and the evacuation of the affected populations as separate problems. This differs from the reality that emergency managers face when having to evacuate an isolated area, since these problems interact with each other. Furthermore, the solution methods presented to solve related previously developed problems cannot simply be re-used and therefore required the design of new solution approaches. Considering the research gaps mentioned in the previous section, and the need for solutions to evacuate isolated communities, the formulation presented in this paper is novel and highly relevant. The contributions of this paper to the research body are as follows.

1. The ICEP is the first study that uses a resource-routing approach to model the evacuation of communities without road-based evacuation routes. It takes into account the specific constraints of isolated communities, where self-evacuation is difficult

and where evacuation resources are mostly heterogeneous in their capabilities, capacities, and compatibilities with potential pick-up and shelter locations;

2. The expansion of the D-ICEP to a two-stage stochastic program with recourse (S-ICEP) allows planning for emergency evacuations by incorporating uncertainty through a set of scenarios;

3. A variety of objective functions for S-ICEP allow the decision makers to prioritize between time and cost efficiency in planning for emergency evacuations;

4. The structure-based local search heuristics presented for the D-ICEP and the S-ICEP allow the problems to be solved efficiently despite their heterogeneous structure with limited penalty on the optimality of the results.

The remainder of this paper is structured as follows: Section 2 provides assumptions about the ICEP. Following that, the deterministic version of the ICEP (D-ICEP) is constructed through a step-wise creation of the required network components in Section 2.2 and its mathematical formulation is introduced in Section 2.3. The D-ICEP is then further analyzed for parameter choices. The stochastic planning problem S-ICEP is introduced in Section 3.2. Different objective functions are also introduced that can be used for different policy implications and evaluated for their effects on the solution provided by the model. Section 4 presents a structure-based, two-phase heuristic to solve the primary objective function of the D-ICEP, including the results of numerical experiments benchmarking the heuristic against a commercial solver. On the basis of the deterministic heuristic, a heuristic to solve the S-ICEP is presented and numerical benchmarking experiments are shown. Lastly, Section 5 provides conclusions and future directions for research.

2. Deterministic problem formulation

2.1. Assumptions

The following assumptions were made to formulate the D-ICEP model.

1. All road connections out of the disaster area are considered disrupted. Therefore, the ability of people to self-evacuate is limited to using private vehicles that do not rely on roads such as aircraft or boats. However, since the majority of people do not own such resources, the majority share of the evacuation requires the use of external resources.

2. The evacuee populations are distributed in between different locations of the affected area.

3. The evacuee population is considered large enough to require a significant amount of resources and/or multiple trips to evacuate.

4. A central planning entity has full authority over planning and coordination of a fleet of recovery resources, except for private modes of transportation.

5. The central planning entity aims to minimize the total evacuation time.

6. All recovery resources considered are located within reasonable distance to the affected area and may differ in their capabilities in terms of their contracting cost, variable operating cost, carrying capacity, loaded and unloaded travel speeds, loading times, time to availability, initial locations, and their compatibility with potential pick-up and drop-off points in the affected area.

7. All recovery resources start from their initial positions once they have been staffed, and travel to a pick-up location in the affected area, and they alternate in between pick-up locations and shelter locations until the number of evacuees is zero, ending at a shelter location.

8. For model simplicity, recovery resources visit only one evacuation pick-up point and one drop-off point per trip.

9. The sets of initial resource positions, evacuation pick-up points, and shelters are known. Shelter and pick-up point identification is thus not part of this problem.

10. The population of evacuees will be at the pick-up locations upon arrival of resources, such that arrival rates of evacuees do not have to be considered. This entails that the evacuees travel to the pick-up locations either by foot or other modes of transport. It should be noted that at this point the transportation of evacuees to the pick-up points is out of scope of this model and needs to be considered separately in future work.

11. Evacuees are considered safe once they have been dropped off at a shelter location.

12. Recovery resources are operating continuously without downtime.

13. Recovery resources are accessible and prepared for all types of evacuees, including children and mobility impaired populations.

14. The capacity of pick-up locations and shelter locations is considered infinite.

2.2. Design of the deterministic ICEP

The D-ICEP minimizes the total evacuation time of a given disaster with fixed evacuee numbers and a fixed set of recovery resources. The network presented in Fig. 1 illustrates the physical flows of evacuees. Let s denote the source node, representing the entire isolated community. The as denote geographically separated evacuation areas, the bs denote the evacuation pick-up points, the cs denote the shelters, and t denotes the sink node. Green arcs indicate routes on which passengers can be transported. Blue arcs show the routes of people who decide to self-evacuate using private vehicles. If it was easy to determine the capacities and transit times for all arcs, the minimal evacuation time could be found through a quickest flow formulation (Burkard et al., 1993).
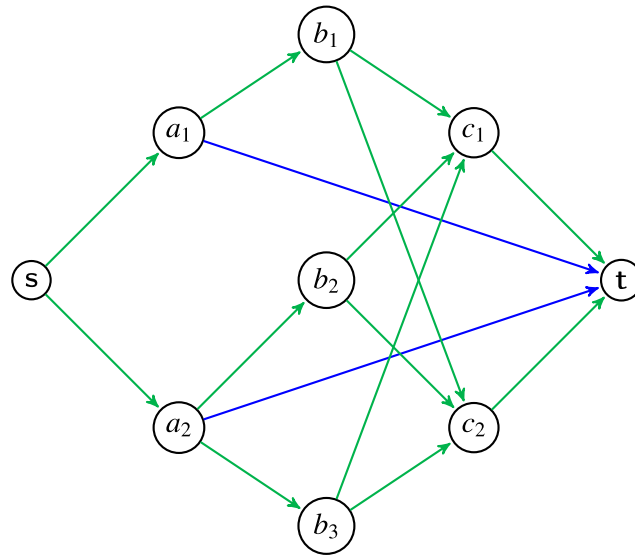
Fig. 1. Illustration of population flows from evacuated to safe locations for two evacuated areas, three evacuation pick-up points, and two shelters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
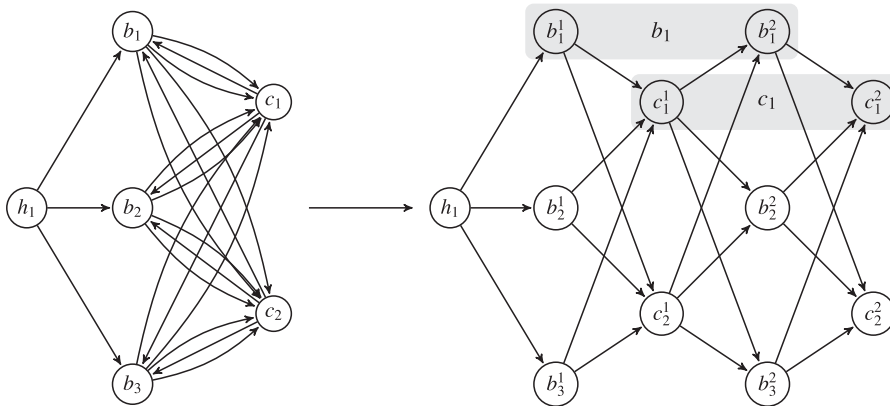

Fig. 2. Illustration of a route network among initial resource location, evacuation areas, and safe areas for two trips to the evacuation location, three evacuation pick-up points, two shelters and two initial resource locations.

However, the capacities and time parameters for the arcs between pick-up points and shelter locations are non-linear because for the ICEP, these depend on which resources are used on a route and how many round trips are made to evacuate the area. Furthermore, resources each have different starting points and may not be compatible with every evacuation pick-up point. For example, a ferry that is in regular service between an island and the mainland might be nearby, but it will only be able to dock at a specifically designed ferry dock. This requires individual routing decisions for each resource and therefore the introduction of binary variables for routing choices, making the problem a mixed-integer formulation. Fig. 2 illustrates the resource routing problem for a single resource. Let h denote the initial resource location. The bs and cs denote the evacuation pick-up points and shelters respectively, as in Fig. 1. A resource travels from its initial location h to a pick-up point b and transports evacuees to a shelter c and returns to one of the compatible evacuation pick-up points b (not necessarily the same as the one it served in the previous trip) for another trip. The left part of Fig. 2 illustrates this problem for a routing problem with one round trip back to the evacuation location and back to the safe location. Breaking down the bs and cs into subnodes for each round trip, as illustrated in the right part of Fig. 2, expands the model structure into a trip-expanded structure, which follows a logic similar to that of the time-expansion presented by Ford and Fulkerson (1958).

With the network presented in Fig. 2, resource routes can be optimized, provided that a separate routing scheme is in place for every considered resource, as the arc capacities and costs in this network still depend on which resource is used on each route. Using this network structure, the arcs can determine the travel time as a function of distance in between nodes and resource speed, while the arcs from b to c also maintain flow capacity according to the corresponding resource carrying capacity. The h nodes further
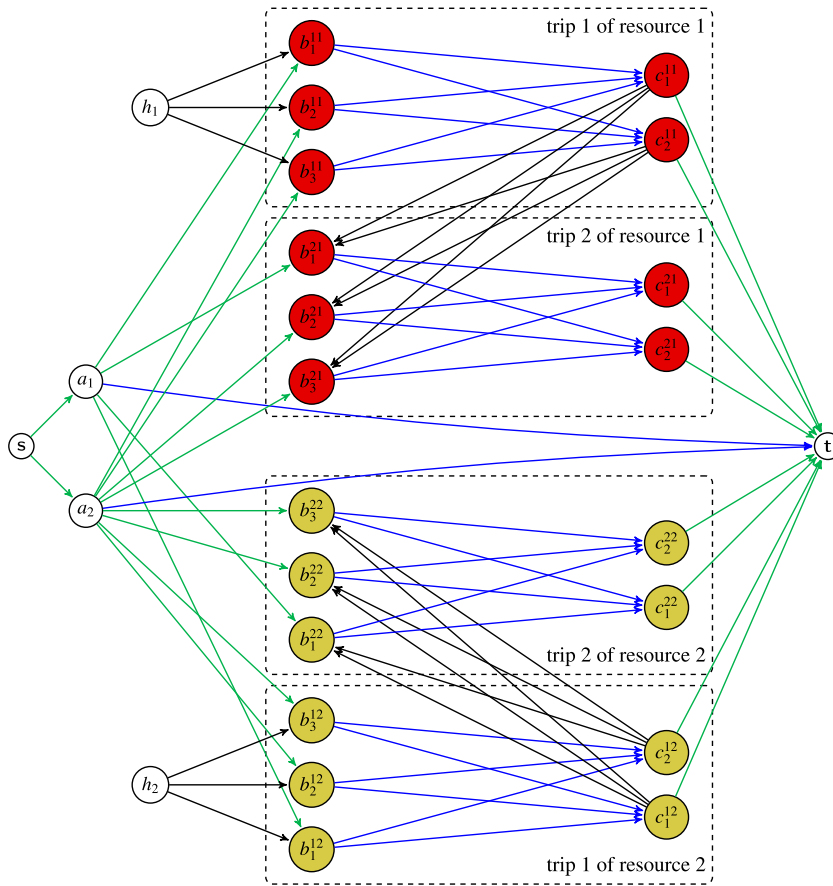
Fig. 3. Illustration of a combined max-flow and routing problem for two trips per resource with three pick-up points, two shelters, and two fully compatible resources. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

determine the time to availability of a resource, and the nodes b and c, the passenger loading and unloading time respectively. Optimizing this sub-network as a shortest path problem would provide the path with the lowest time consumption.

Integrating the passenger flows from Fig. 1 with the routing network from Fig. 2 for multiple resources produces the network visualized in Fig. 3. Arcs with no flows are visualized in black, arcs with finite capacity in blue, and arcs with infinite capacity in green. Each node corresponding to a specific resource is illustrated in a different color and the boxes show the frame of a round trip for a resource. The remaining parts of the network work as described previously in Figs. 1 and 2. Note that the network as illustrated in Fig. 3 assumes full compatibility between resources and evacuation pick-up points and shelters. For limited compatibility, arcs in between nodes and resources that are incompatible have to be removed from the network. This network defines D-ICEP, and solving it with the objective of minimal route completion time generates an optimal evacuation route plan for the isolated community the provided data represents.

The D-ICEP can be considered a trip-expanded heterogeneous fleet variant of the location routing problem (LRP) with multiple node visits, where the total route plan length is minimized. Since the LRP, a generalization of the VRP, is NP-hard, so is the D-ICEP. The problem size and the required computational run time to find an optimal solution therefore increase exponentially when instances are added to the problem sets. A particular structural challenge in solving this problem is the heterogeneity and limited compatibility of the resource set. This makes the solution space more complex than for a symmetric resource set, leading to a higher risk of difficulties in solution discrimination for the solver. It also makes it more difficult to modify an existing solution through existing local search methods, as the routes in between different resources are not fully interchangeable. These challenges are also heavily influenced by the provided data set. Depending on how compatible the resources considered are with the pick-up and drop-off points, the number of route options varies a lot. A fully compatible resource set therefore has a lot more flexibility than a set with limited compatibility in choosing routes, but also requires higher computational effort. The following section describes the D-ICEP in mathematical terms. Table 2 introduces the notation for the formulation, followed by the problem formulation.

Table 2
Notation key for D-ICEP.

Sets	Set description	Parameters	Parameter description
$i \in I$	Recovery resources	q_i	Passenger capacity of resource i
$k \in K$	Potential round trips per resource	u_i	Time to availability of resource i
s	Source node	o_i	Loading time of resource i
$a \in A$	Evacuation areas	p_i	Unloading time of resource i
$b \in B$	Pick-up points in evacuation area	d_a	Evacuation demand at location a
$c \in C$	Drop-off points in safe locations	g_a	Max. no. of self-evacuations from area a
t	Sink node	t_{hb}^i	$\frac{distance(h \rightarrow b)}{\text{empty travel speed of resource } i}$: cost of arc ζ_{hb}^{1i}
$h \in H$	Initial resource locations	t_{bc}^i	$\frac{distance(b \rightarrow c)}{\text{loaded travel speed of resource } i}$: cost of arc γ_{bc}^{ki}
		t_{cb}^i	$\frac{distance(c \rightarrow b)}{\text{empty travel speed of resource } i}$: cost of arc δ_{cb}^{ki}, Only $k = 1, \dots, K-1$

Arcs	Arc description	Variables	Variable description
$\alpha_{sa} \in \bar{A}$	Source s to area a	fl_{at}	Flow on arc λ_{at}
$\beta_{ab}^{ki} \in \bar{B}$	Area a to pick-up b of trip k for resource i	fl_{ab}^{ki}	Flow on arc β_{ab}^{ki}
$\gamma_{bc}^{ki} \in \bar{\Gamma}$	Pick-up b to drop-off c of trip k for resource i	fl_{bc}^{ki}	Flow on arc γ_{bc}^{ki}
$\delta_{cb}^{ki} \in \bar{\Delta}$	Drop-off c to pick-up b of trip k to trip $k+1$	fl_{ct}^{ki}	Flow on arc ϵ_{ct}^{ki}
	For resource i, for $k = 1, \dots, K-1$	w_{hb}^{1i}	{**1**: if route on ζ_{hb}^{1i} selected, **0**: otherwise}
$\epsilon_{ct} \in \bar{E}$	Drop-off c to sink node t	x_{bc}^{ki}	{**1**: if route on γ_{bc}^{ki} selected, **0**: otherwise}
$\zeta_{hb}^{1i} \in \bar{Z}$	Initial resource location h to pick-up b	y_{cb}^{ki}	{**1**: if route on δ_{cb}^{ki} selected, **0**: otherwise}
	For resource i, on trip 1	r	Total evacuation time
$\lambda_{at} \in \bar{\Lambda}$	Area a to sink node t, for private evacuations	s_i	Route completion time of resource i

2.3. Deterministic problem formulation (D-ICEP)

2.3.1. D-ICEP formulation

$$\min \quad r \tag{1}$$

$$s.t. \quad r \geq s_i \qquad \forall i \in I \tag{2}$$

$$s_i = \sum_{\zeta_{hb}^{1i} \in \bar{Z}} \left(t_{hb}^i w_{hb}^{1i} \right) + \sum_{\gamma_{bc}^{ki} \in \bar{\Gamma}} \left(t_{bc}^i x_{bc}^{ki} \right) + \sum_{\delta_{cb}^{ki} \in \bar{\Delta}} \left(t_{cb}^i y_{cb}^{ki} \right) +$$
$$\sum_{\zeta_{hb}^{1i} \in \bar{Z}} \left(u_i w_{hb}^{1i} \right) + \sum_{\zeta_{hb}^{1i} \in \bar{Z}} \left(o_i w_{hb}^{1i} \right) +$$
$$\sum_{\delta_{cb}^{ki} \in \bar{\Delta}} \left(o_i y_{cb}^{ki} \right) + \sum_{\gamma_{bc}^{ki} \in \bar{\Gamma}} \left(p_i x_{bc}^{ki} \right) \qquad \forall i \in I \tag{3}$$

$$fl_{at} \leq g_a \qquad \forall \lambda_{at} \in \bar{\Lambda} \tag{4}$$

$$fl_{bc}^{ki} \leq q_i(x_{bc}^{ki}) \qquad \forall \gamma_{bc}^{ki} \in \bar{\Gamma} \tag{5}$$

$$d_a = fl_{at} + \sum_{\beta_{jb}^{ki} \in \bar{B}: j=a} fl_{ab}^{ki} \qquad \forall a \in A \tag{6}$$

$$\sum_{\beta_{aj}^{ki} \in \bar{B}: j=b} fl_{ab}^{ki} = \sum_{\gamma_{jc}^{ki} \in \bar{\Gamma}: j=b} fl_{bc}^{ki} \qquad \forall b \in B, \forall k \in K, \forall i \in I \tag{7}$$

$$\sum_{\gamma_{bj}^{ki} \in \bar{\Gamma}: j=c} fl_{bc}^{ki} = fl_{ct}^{ki} \qquad \forall c \in C, \forall k \in K, \forall i \in I \tag{8}$$

$$\sum_{\zeta_{hb}^{1i} \in \bar{Z}} w_{hb}^{1i} \leq 1 \qquad \forall i \in I \tag{9}$$

$$\sum_{\gamma_{bc}^{ki} \in \bar{\Gamma}} x_{bc}^{ki} \leq 1 \qquad \forall i \in I, k \in K \tag{10}$$

$$\sum_{\delta_{cb}^{ki} \in \bar{\Delta}} y_{cb}^{ki} \leq 1 \qquad \forall i \in I, k \in K \setminus \{k = K\} \tag{11}$$

$$\sum_{h \in H} w_{hb}^{1i} = \sum_{c \in C} x_{bc}^{1i} \qquad \forall b \in B, \forall i \in I \tag{12}$$

$$\sum_{c \in C} y_{cb}^{(k-1)i} = \sum_{c \in C} x_{bc}^{ki} \qquad \forall b \in B, \forall i \in I, \forall k \in K \setminus \{k = 1\} \tag{13}$$

$$\sum_{b\in B} x_{bc}^{ki} \geq \sum_{b\in B} y_{cb}^{ki} \qquad \forall c \in C, \forall i \in I, \forall k \in K \setminus \{k = K\} \qquad (14)$$

$$fl_{at} \geq 0 \qquad \forall \lambda_{at} \in \bar{\Lambda} \qquad (15)$$

$$fl_{ab}^{ki} \geq 0 \qquad \forall \beta_{ab}^{ki} \in \bar{B} \qquad (16)$$

$$fl_{bc}^{ki} \geq 0 \qquad \forall \gamma_{bc}^{ki} \in \bar{\Gamma} \qquad (17)$$

$$fl_{ct}^{ki} \geq 0 \qquad \forall \epsilon_{ct}^{ki} \in \bar{E} \qquad (18)$$

$$s_i \geq 0 \qquad \forall i \in I \qquad (19)$$

$$r \geq 0 \qquad (20)$$

$$w_{hb}^{1i} \in \{0, 1\} \qquad \forall \zeta_{hb}^{1i} \in \bar{Z} \qquad (21)$$

$$x_{bc}^{ki} \in \{0, 1\} \qquad \forall \gamma_{bc}^{ki} \in \bar{\Gamma} \qquad (22)$$

$$y_{cb}^{ki} \in \{0, 1\} \qquad \forall \delta_{cb}^{ki} \in \bar{\Delta} \qquad (23)$$

The D-ICEP minimizes the total evacuation time r. The time constraint (2) lower bounds r with the highest route completion time of any resource, which is defined in (3). Capacity constraint (4) ensures that no more private self-evacuations can occur than denoted per location a. (5) ensures that the arc capacity is limited by the capacity of the corresponding resource, if the arc is selected as part of the resource route (5), and if no resource is selected, to be zero. Flow conservation constraints are (6) through (8), which ensure that the inflow of evacuees equals the outflows at every node except the source, sink and initial resource location nodes. Constraints (9) through (11) ensure that a maximum of one connection per route segment can be selected for each resource at a time. Route adjacency constraints (12) through (14) ensure that on every leg of a trip, the resources depart from the same node they arrived at on the previous leg. Route adjacency constraint (12) secures this for the arrival from the initial resource location and constraint (13) does this for all other round trips. Ultimately, route adjacency constraint (14) ensures that a resource does not have to return to an evacuation location if no potential evacuees are left. Lastly, variables (15) through (18) define all flows as non-negative continuous variables, variables (19) and (20) define the time-related variables as non-negative continuous, variables (21) through (23) define all route selections as binary variables.

2.3.2. Considerations on the number of round trips

The resource set I, the number of evacuation areas A, pick-up points B, shelter drop-offs C, and initial resource locations H are usually fixed. Because of the heterogeneity of the resources, increasing the maximum number of round trips K per resource i by one trip, increases the size of the problem by the sum of pick-up and drop-off points multiplied by the number of resources. Therefore, choosing arbitrarily large sets will inflate the size of the problem and the computational effort, while not improving the results. However, if the resource and round trip sets are unreasonably small, the problem will cause a penalty for D-ICEP, since only a part of the population can be evacuated. A lower bound to the required size of the resource set I and the number of round trips K can be found if the requirement presented in (24) is met.

$$K \sum_{i\in I} q_i \geq \sum_{a\in A} d_a \qquad (24)$$

This requirement is simple to obtain. However, it would only work for a single solution that does evacuate the entire population, that is only if every resource i would do exactly K round trips and if the full capacity of every resource i can be used on every trip k. This can be far from optimal and may not even be feasible in reality. However, it is not possible to determine by how much to increase K exactly without solving the D-ICEP problem by itself. It is therefore recommended to choose the number of round trips with a good safety margin in the above equation on the right hand side of the equation (RHS), e.g. through $K \sum_{i\in I} q_i \geq 2 \sum_{a\in A} d_a$. On the other end of the spectrum, an upper bound for the number of round trips can be calculated through the formula presented in (25). The maximum number of trips can be determined by the entire evacuation demand divided by the capacity of the resource with the smallest capacity, which represents the case if only this resource would have to complete the entire population from the area.

$$\max \ K \leq \left\lceil \frac{\sum_{a\in A} d_a}{q_i} \right\rceil \quad \{i = \arg\min q_i\} \qquad (25)$$

3. Stochastic problem formulation

3.1. Additional assumptions

As presented above, the D-ICEP problem can be used for response purposes, particularly when all disaster parameters are known. For planning purposes, securing an optimal set of recovery resources for a possible disaster is crucial. However, a planner needs to prepare the resources for a variety of disaster scenarios, since the exact nature of a potential disaster is unknown. Therefore, in the following sections, the problem is expanded to a stochastic problem and the S-ICEP formulation is introduced. The following assumptions in addition to the ones presented for D-ICEP have to be made for S-ICEP.

1. The exact disaster nature and time of occurrence is unknown.
2. The evacuation population in the area is uncertain because of seasonal fluctuation.
3. Weather conditions can influence travel times for resources.
4. The planning authority aims to select an optimal set of resources that minimizes expected evacuation time for a variety of disaster scenarios, while considering cost efficiency.

3.2. Expansion to a stochastic problem

During the planning stage, it is not possible to know how a disaster will turn out and where and how many people have to be evacuated. The underlying uncertainty affects a variety of factors: which resources are actually useful, the route choices the resources have to take, how many trips each of the resources have to complete, and ultimately what the resulting evacuation time is. Based on the D-ICEP, emergency planners want to optimally select a fleet of recovery resources that can handle a variety of evacuation scenarios, given the ability of a resource to reduce evacuation time and offer competitive cost structures. The model takes into account the costs of choosing a resource as part of the fleet, variable operating costs, such as labor and fuel, as well as the calculated cost of the loss of a human life. This needs to be considered to avoid minimizing evacuation time at the cost of not evacuating the entire population. Each evacuation resource needs to be determined before a disaster occurs based on its capabilities, given uncertainty about the location, number of evacuees, and the weather conditions at the time of the disaster. Typical optimization approaches to incorporate uncertainty are:

1. Evaluate the uncertainty by constructing an expected-value problem
2. (Multi-stage) stochastic dynamic programming
3. Robust optimization
4. Two-stage stochastic programs with recourse

The results from deterministic models oversimplify the model since they do not consider variety in the disaster nature. Further, the complexity of using stages of evolving knowledge that require stochastic dynamic programming solutions (Bellman, 1966) is not considered to be necessary for the version of the S-ICEP presented in this paper. Therefore, these two methods are not ideal for the S-ICEP. While robust optimization offers advantages by providing a robust solution (Soyster, 1973; Gorissen et al., 2015), the high uncertainty over the disaster nature considered for the ICEP could lead to overly conservative solutions and thus lead to a much larger resource set than actually needed. Two-stage stochastic programs with recourse (Shapiro, 2008), on the other hand, offer the flexibility to design problems under uncertainty distributions. As an alternative to parameter distributions, two-stage stochastic programs also allow for the design of specific, probability-weighted scenarios. Scenario-based planning techniques can help first responders to clearly separate different disaster outcomes, for example between evacuations during different seasons of the year. With stochastic recourse models, differences among scenarios can be simply accounted for by using scenario specific travel distance and incidence matrices. Therefore, two-stage stochastic programs with recourse are a good fit for S-ICEP.

When expanding the D-ICEP into a two-stage stochastic programming framework, the resource fleet selection needs to be scenario-independent and, hence, needs to be decided in the first stage of the model. The second stage of the model must then find the optimal route plan for each scenario of interest, given the resource fleet selected in the first stage. Using the recourse component, the S-ICEP can select the resource fleet that provides minimal expected evacuation time over multiple scenarios, while considering cost imposed by the resource configuration. This results in a model structure that solves the D-ICEP in its second stage for each scenario of interest. Once the disaster happens, and the uncertainty is revealed, the D-ICEP has to be solved, with the resource set fixed to the decision made in the first stage of the problem. Since the S-ICEP contains the D-ICEP in its second stage, the challenges and parameter considerations of the D-ICEP also apply to the S-ICEP. This also applies to the complexity of the problem, which is therefore also NP-hard.

The following section introduces the S-ICEP as an expansion of the D-ICEP. Therefore, only additional required sets, parameters, variables, and constraints that are not already in D-ICEP are introduced in Table 3.

3.3. Stochastic problem formulation (S-ICEP)

3.3.1. Problem formulation

$$\min \quad \frac{\sum_{i \in I} cf_i(z_i)}{\sum_{i \in I}(cf_i + cv_i(T))} + \mathbb{E}[C(z, \xi)] \tag{26}$$

$$s.t. \quad z_i \in \{0, 1\} \qquad\qquad \forall i \in I \tag{27}$$

where

$$C(z, \xi) := \min \quad r + \frac{\sum_{i \in I} cv_i(s_i)}{\sum_{i \in I}(cf_i + cv_i(T))} + P \sum_{a \in A} n_a \tag{28}$$

$$s.t. \quad D\text{-}ICEP \text{ constraints } (2)\text{–}(5), (7)\text{–}(8), (12)\text{–}(23),$$

$$r \le T \tag{29}$$

Table 3
Additional notations for S-ICEP (in addition to D-ICEP)

Sets	Set description
$\xi \in \varXi$	set of evacuation scenarios passed to the model
Parameters	**Parameter description**
cf_i	Fixed cost parameter of selecting resource i into the evacuation fleet
cv_i	Variable cost parameter for resource i
T	Upper time limit for total evacuation time r
P	Penalty cost for each person that is not successfully evacuated
Variables	**Variable description**
z_i	{**1**: if resource i gets selected into resource fleet, **0**: otherwise}
n_a	Number of non-evacuated people at area a

$$fl_{bc}^{ki} \le q_i(z_i) \qquad\qquad \forall \gamma_{bc}^{ki} \in \bar{\varGamma} \qquad (30)$$

$$d_a(\xi) = fl_{at} + \sum_{\beta_{jb}^{ki} \in \bar{B}:\, j=a} fl_{ab}^{ki} + n_a \qquad\qquad \forall a \in A \qquad (31)$$

$$\sum_{\zeta_{hb}^{1i} \in \bar{Z}} w_{hb}^{1i} \le z_i \qquad\qquad \forall i \in I \qquad (32)$$

$$\sum_{\gamma_{bc}^{ki} \in \bar{\varGamma}} x_{bc}^{ki} \le z_i \qquad\qquad \forall i \in I, k \in K \qquad (33)$$

$$\sum_{\delta_{cb}^{ki} \in \bar{\varDelta}} y_{cb}^{ki} \le z_i \qquad\qquad \forall i \in I, k \in K \setminus \{k = K\} \qquad (34)$$

$$n_a \ge 0 \qquad\qquad \forall a \in A \qquad (35)$$

The S-ICEP objective function (26) combines multiple objectives. It aims to minimize the expected evacuation time ($r(\xi)$), the number of people not evacuated ($P\left(\sum_{a \in A} n_a(\xi)\right)$), and the total cost of the evacuation plan ($\sum_{i \in I} cf_i(z_i) / \sum_{i \in I}(cf_i + cv_i(T))$) and ($\sum_{i \in I} cv_i(s_i(\xi)) / \sum_{i \in I}(cf_i + cv_i(T))$). The first stage decision variable (27) determines the set of resources that will be used for evacuation, which fixes the fixed-cost component of the evacuation plan cost. The uncertain second-stage determines the evacuation time for every provided scenario given the evacuation fleet from the first stage. The second stage is essentially a slightly modified version of the D-ICEP, with the objective consisting of the total evacuation time, as in the D-ICEP, plus the variable cost and a penalty that is applied for every person that could not be evacuated. This allows for not evacuating the entire population if a scenario is extreme and has a low probability. The ranking of objective components is ensured through normalizing the cost components, such that their maximum influence on the objective function is 1. This ensures that an improvement by at least one time unit in evacuation time will always dominate the cost objective so that evacuation speed is prioritized over cost. It is left to the modeler to decide the granularity of time units – minutes or even seconds – considered in the objective. Restricting the influence of the evacuation cost in this way is based on Sherali's (Sherali, 1982) approaches for lexicographic multi-objective functions. This objective function further includes a penalty that is applied for every person that could not be evacuated, which is modeled through the added decision variable n_a, as indicated in constraint (35). In combination with replacing constraint (6) with constraint (31), this ensures that the problem will still provide an evacuation plan, even if parameter T is set so low that not everyone can be evacuated. Decision makers can adjust the size of P to control the desired risk level for not evacuating everyone in extreme scenarios. The remaining constraints of the second stage are almost the same as the constraints of the D-ICEP with a few modifications. As mentioned above, an upper time limit parameter was added (29). Constraint (30) was added to ensure that flows can only be allocated to a resource route if the resource was also selected in the first stage. Furthermore, for D-ICEP constraints (9), (10), and (11) the RHS was replaced by z_i as shown in constraints (32) through (34), to make sure that routes can only be connected if the resource was selected in the first stage. Otherwise the S-ICEP constraints are equivalent to the D-ICEP constraints. In the following section, the S-ICEP is analyzed for its structural challenges, sensitivity to set sizes, objective functions, and parameter choices, and their effects on the feasibility of the model. Since the D-ICEP is essentially a subset of the S-ICEP, the same findings also apply to the D-ICEP, with the exception of the differences in objective functions, scenario design, and resource sets.

3.3.2. Scenarios

For the S-ICEP, using a set of reasonably varied and realistic scenarios is the main challenge, as it is crucial for obtaining meaningful results. The S-ICEP model provides the modeler with flexibility in determining which parameters they want to modify. A modeler could, for example, investigate the effects of seasonal population fluctuation on the evacuation time for one specific disaster case. Alternatively, for areas with stable populations, a modeler could investigate the effects on different affected populations. Furthermore, differences in weather patterns that influence travel times or pick-up and drop-off point access can be modeled.

As mentioned in Section 3.2, once the uncertainty that the S-ICEP considers during the planning process is resolved, emergency planners face a D-ICEP, which may be different from all the scenarios considered. The goal of the S-ICEP, therefore, is to provide a resource set that can perform well in case of an actual disaster. Therefore, the number and variety of scenarios should be chosen in

a way that approximates the underlying uncertainties without overfitting the solution. If sufficient data is available for the region of interest, this is how scenarios should be designed. The challenge with planning for disasters is that sufficiently detailed data on where a disaster could occur and how it would evolve is often difficult to obtain. In some cases, specific scenarios can be obtained through collaboration with experts, such as experienced first responders or emergency planners, who can rely on experience to identify where disasters are likely to occur and can evaluate the relative probability of the scenarios. To make the results robust, the modeler should ensure that scenarios cover a variety of realistic cases that may include differences in the total number of evacuees, the number of evacuation areas affected, and their distribution in between the evacuation areas. Differences in weather patterns that may affect the accessibility of certain pick-up and drop-off points should also be considered, as well as differences in the number of evacuees that self-evacuate. Therefore, a mix of sophisticated data sets (e.g. census data for population estimates) and subject matter expert inputs can be used to design realistic scenarios. However, every additional scenario increases the complexity of the model, which is why the right balance between accurately representing the underlying uncertainties and keeping the number of scenarios limited is important.

3.3.3. Resource sets

Section 2.3.2 already discussed the importance of setting parameter K. The S-ICEP has an additional degree of freedom through modifying the considered set of resources I. While determining the optimal set of resources is the objective of the S-ICEP, the size of the potential resource set I also inflates the problem size. Table 8 illustrates the effect on the number of variables. The bounds from Eqs. (24) and (25) should be considered when determining the size of the potential resource set.

3.3.4. Evacuation time limits

T is provided to allow emergency planners to set a desired maximum evacuation time. Reviewing the S-ICEP formulation from Section 3.3.1, it is easy to verify that an unreasonably small T will cause the resulting evacuation plan to not evacuate the entire population in some or all scenarios. Instead of simply letting the model return infeasible in this case, this modeling choice provides the emergency planner with additional information by how much the target T was missed, since it is indicated through the number of non-evacuated people. An emergency modeler may then consider requesting more evacuation resources in the area. However, setting T too high also has downsides from a computational perspective. Through setting T, the solution space can be significantly restricted. However, finding a feasible lower bound to T is not trivial, because it can only be obtained by solving the S-ICEP problem and is essentially the goal of the problem.

$$\min \ T = \min \ r \ (s.t. \ \text{ICEP constraints}) \tag{36}$$

However, this property can be useful during the emergency planning process. Through experimenting with S-ICEP, modelers can find out how much time a route plan is expected to take. They can then use this knowledge to add constraint (29) to the D-ICEP formulation and set a parameter T in the D-ICEP, thus restricting the solution space of D-ICEP and helping the solver to find the optimal solution more quickly. This is further investigated in Section 3.4.5 with regards to solution time. An upper bound to the time limit can be derived by maximizing r, although this is not of much use in solving the problem.

3.3.5. Penalty parameters

The choice of the penalty parameter P applied to the S-ICEP strongly affects the provided policy. This penalty parameter is mostly an applied measure of calculated risk in the design of the evacuation plan instead of the true cost of not evacuating a person from the affected area. It is supposed to dominate the other terms in the objective function and if chosen too small, solutions may be favored that evacuate a smaller population than possible because it might be comparably cheaper to leave people in the affected area than conducting an additional round trip with a subset of resources. Similarly, an unreasonably large choice of P may result in an impractical rule that no person can be left behind in any scenario, no matter how unlikely and extreme that scenario might be (e.g. a scenario with relative probability 0.5%, since the effect on the objective function is still very high even though the penalty is discounted by the probability). A lower bound that ensures dominance over the time and cost components can be established as in (37).

$$P \geq T + \sum_{i \in I} \left(c f_i + c v_i(T) \right) \tag{37}$$

There are caveats to using the penalty parameter as introduced in Section 3.3. It is not possible to control how many people have to be evacuated at minimum. Compared to requiring everyone be evacuated, the penalty parameter also increases computational run time as the solution space increases. To find the right balance, the modeler could determine what percentage of the population should be guaranteed to be evacuated in any scenario and add constraint (38) to the second stage of the S-ICEP model formulation, where m equals the fraction of the population that is guaranteed to be evacuated in every scenario.

$$(1 - m) \sum_{a \in A} d_a \geq \sum_{a \in A} n_a \tag{38}$$

To further control for how many scenarios are allowed to not evacuate the entire population, a chance constraint could be added in the first stage of the S-ICEP. Depending on the desired probability of complete evacuation e, constraint (39) would model this. However, this could further complicate solving the problem as it would introduce a non-linearity.

$$Pr \left(\sum_{a \in A} n_a(\xi) = 0 \right) \geq e, \qquad \forall \xi \in \Xi \tag{39}$$

3.4. Objective functions for S-ICEP

3.4.1. Balanced objective functions

The primary objective function of the S-ICEP used in Section 3.3, further denoted as *Bal_1*, is a multi-objective formulation that prioritizes evacuation time over cost. For evacuation purposes, this is a reasonable balance. Instead of focusing on minimizing the most time consuming route, an alternative formulation aims to minimize the overall sum of route times to generate higher efficiency in individual route choices. This revised balanced objective function was denoted as *Bal_2* (40).

$$\frac{\sum_{i\in I} cf_i z_i}{\sum_{i\in I}(cf_i + cv_i(T))} + \mathbb{E}\left[\sum_{i\in I} s_i + \frac{\sum_{i\in I} cv_i s_i}{\sum_{i\in I}(cf_i + cv_i(T))} + P\sum_{a\in A} n_a\right] \tag{40}$$

3.4.2. Conservative objective functions

More conservative but simpler objective functions can be considered that solely aim to minimize the expected evacuation time, ignoring the fixed and variable cost imposed by the resource usage. Depending on whether to minimize for r or for $\sum_{i\in I} s_i$, these can optimize total evacuation time or the sum of all route times. Eq. (41) denotes *Cons_1*. Alternatively, Eq. (42) provides *Cons_2*.

$$\min \ \mathbb{E}\left[r + P\sum_{a\in A} n_a\right] \tag{41}$$

$$\min \ \mathbb{E}\left[\sum_{i\in I} s_i + P\sum_{a\in A} n_a\right] \tag{42}$$

3.4.3. Economic objective functions

Economic objective functions can also be considered that minimize the expected evacuation cost. The expected variable cost is calculated as the sum of each variable cost rate multiplied by the time consumption of each selected route segment over all selected resources, plus the penalty cost for leaving a person behind. This objective therefore automatically ensures cost efficient route choices. This objective function was denoted *Econ_1* (43).

$$\min \ \sum_{i\in I} cf_i z_i + \mathbb{E}\left[\sum_{i\in I} cv_i s_i + P\sum_{a\in A} n_a\right] \tag{43}$$

If *Econ_1* is considered to be too budget focused and if a non-dominant incorporation of evacuation time is desired, then the total evacuation time can be discounted by its upper bound, as shown in the multi-objective objective function (44), which is denoted as *Econ_2*.

$$\min \ \sum_{i\in I} cf_i z_i + \mathbb{E}\left[\sum_{i\in I} cv_i s_i + \frac{r}{T} + P\sum_{a\in A} n_a\right] \tag{44}$$

3.4.4. Discretization of objective functions

All objective functions can further be discretized into a deterministic equivalent if the number of scenarios is finite. Eq. (45) provides an example for the objective function *Bal_1* with two scenarios with probabilities p_1 and p_2, where $\sum_{\xi\in\Xi} p_\xi = 1$.

$$\min \ \frac{\sum_{i\in I} cf_i(z_i)}{\sum_{i\in I}(cf_i + cv_i(T))} + p_1\left(r(\xi_1) + \frac{\sum_{i\in I} cv_i(s_i(\xi_1))}{\sum_{i\in I}(cf_i + cv_i(T))} + P\sum_{a\in A} n_a(\xi_1)\right) +$$
$$p_2\left(r(\xi_2) + \frac{\sum_{i\in I} cv_i(s_i(\xi_2))}{\sum_{i\in I}(cf_i + cv_i(T))} + P\sum_{a\in A} n_a(\xi_2)\right) \tag{45}$$

3.4.5. Effects of S-ICEP objective functions

To investigate the effect of the objective functions further, the three test data sets presented in Table 4 were used to investigate model sensitivity. All computational runs were made on a Mac with a 2.6 GHz Dual-Core Intel Core i5 CPU, using an implementation of S-ICEP in the Pyomo interface for Python on the Gurobi 9.0 commercial solver, with a run-time limit of 3600 s As recommended by the Gurobi environment, the deterministic equivalent of the S-ICEP was solved. The settings chosen for Gurobi 9.0 were to solve the problem using the root node model of the MIP as it delivered the best performance. Using the concurrent version of the solver did not provide improvements in run time for this model.

By applying the objective functions introduced in Section 3.4 to the data sets from Table 4, the results displayed in Table 5 were obtained. The table presents the key parameters of the solution for each data set and each objective function, and the expected evacuation time for each scenario.

Table 5 shows that for all data sets, conservative objective functions generally led to short evacuation times. The solutions provided by the balanced objective functions provided almost the same solutions with regards to total evacuation time, but with more efficient resource choices. This shows that adding the cost component to the objective function helps in reducing expected cost while maintaining quick evacuation plans, though the *Bal_1* objective, that includes the minimization of the total evacuation time, was more reliable. The results for data set $I2$ show that *Bal_2* does not lead to the same type of evacuation time as *Cons_2*, although both include the sum of route times objective. It can furthermore be observed that objective functions that minimized the sum of

Table 4
Test data sets for Objective Function Evaluation.

Sets	I1	I2	I3
	Set size		
Scenarios	2	3	4
Potential resources	6	6	8
Initial storage locations	1	2	3
Evacuation locations	3	4	5
Evacuation pick-up points	5	6	8
Safe drop-off points	2	3	3
Round trips	6	8	8
Parameters	Setting		
Penalty	5,000	5,000	5,000
Evacuation time limit (min)	120	240	500
Variable type	Quantity		
Continuous variables	832	2,718	6,640
Binary variables	858	3,328	8,352

Table 5
Results for different objective functions.

Data set	Objective	Objective	Exp. time (min)	Cost ($)	Run-time	Resources
I1	$Cons\_1$	100.32	(72, 119)	5,671.67	2.05 s	6/6
	$Cons\_2$	327.04	(90, 120)	5,552.13	0.93 s	6/6
	Bal_1	101.83	(72, 119)	5,671.67	5.15 s	6/6
	Bal_2	327.98	(90, 120)	5,052.13	0.75 s	5/6
	$Econ\_1$	2,955.64	(102, 120)	2,955.64	1.09 s	4/6
	$Econ\_2$	2,956.58	(102, 120)	2,956.58	1.25 s	4/6
I2	$Cons\_1$	118.22	(132, 159, 82)	6,255.75	31.14 s	6/6
	$Cons\_2$	233.62	(232, 190, 82)	5,573.58	1.71 s	6/6
	Bal_1	119.6	(132, 159, 82)	5,155.77	725.32 s	5/6
	Bal_2	234.56	(232, 229, 82)	5,073.58	4.30 s	5/6
	$Econ\_1$	2,522.19	(228, 228, 226)	2,522.19	48.13 s	3/6
	$Econ\_2$	2,522.19	(219, 229, 169)	2,522.19	29.8 s	3/6
I3	$Cons\_1$	117.72	(132, 313, 88, 91)	6,989.66	7.80 s	8/8
	$Cons\_2$	290.92	(282, 495, 142, 182)	6,384.41	3.88 s	8/8
	Bal_1	118.82	(132, 313, 88, 91)	6,821.52	3,600.00 s[a]	8/8
	Bal_2	291.79	(282, 495, 142, 182)	5,384.41	4.36 s	6/8
	$Econ\_1$	2,614.04	(374, 495, 321, 317)	2,614.04	61.34 s	4/8
	$Econ\_2$	2,614.04	(374, 495, 270, 242)	2,614.04	6.52 s	4/8

[a]Results were aborted after 3600 s; the best available solution is displayed.

route times ($Cons\_2$, Bal_2) did not provide the same solution quality with regards to total evacuation time. In fact, minimizing the sum of route times produced results that were closer to solutions of economic objective functions, since the variable cost term was a function of time. Economic objective functions can find reasonably quick solutions, but only if decision makers define tight upper time limits. This is visible in the results for $I1$, where the upper time limit T was set to 120, which is close to the lowest feasible time of 119 of the second scenario. Modelers should therefore carefully consider their priorities when applying these functions to the problem and consider the settings of parameter T.

In addition, any objective function that involved minimizing the total evacuation time instead of the sum of route times showed a significantly larger computational run-time, particularly for $I3$. This indicates the commercial solver's solution discrimination was more difficult when minimizing the total evacuation time. The effect appears amplified by how far away T is set from the optimal solution of each scenario. This is illustrated by the differences between Bal_1 and Bal_2. For data set $I1$, the second scenario had a minimum total evacuation time of 119, but the upper time limit was set to 120, which corresponds to just 0.8% above the optimal solution. Here, Bal_1 showed a run time approximately 6.9 times as high as Bal_2. In $I2$, this factor is increased to approximately 168.7. The time limit was set to 240, which is 81 (50.9%) more than the longest minimum total evacuation time reached for this scenario. In $I3$, while the run for Bal_1 was aborted at $3600s$, the factor is already 825.7 times the run time of Bal_2. In this case, T was set to 500, which is 187 (59.7%) above the longest minimum evacuation time for this scenario.

3.4.6. Learnings from experiments

This illustrates two main findings: the difficulties of the solver to perform effective solution discrimination for minimum evacuation time objective functions, and the sensitivity of the solver to the setting of parameter T in comparison to the minimum total evacuation time for these functions. This makes these objective functions particularly challenging to use for large problems, as the problem cannot easily be decomposed into a problem for each resource. This gives the modeler multiple options when aiming to reduce the computational run time of Bal_1 when it is unclear how to set T:

1. Start with low settings for T and perform algorithm runs, which will likely result in high penalties due to people left behind but short computational run times. Based on the results, gradually increase T and re-run the algorithm until a solution with no one left behind can be obtained.
2. Start with any setting for T and run Bal_2 instead of Bal_1 and gradually decrease T and perform additional runs until a plan is returned that leaves people behind. Choose the previous setting for T and run again with Bal_1. This should return a reasonably short run time for Bal_1 as the gap between the minimum total evacuation time and T should be small enough. This approach takes advantage of the fact that the minimum possible total evacuation time using Bal_2 is equivalent to the optimal solution of Bal_1. Thus if $T = \min r$, Bal_2 would deliver the same solution as Bal_1.
3. Consider alternative approximate solution methods, such as heuristics, metaheuristics and decomposition methods.

This has further implications on how to solve D-ICEP during emergency response. If planing with S-ICEP has been performed and a variety of realistic scenarios have been considered, a estimate on a reasonable upper time limit may have been achieved. The D-ICEP can then be executed during an emergency response situation with T added as an upper time limit, thus accelerating the solution time. Another strategy is to use the solution for the scenario obtained from S-ICEP that is closest to the situation D-ICEP faces, and provide it as a warm start to the solver. Considering that particularly the primary multi-objective function of S-ICEP (Bal_1) is challenging to solve with a commercial solver in a timely manner, details on a heuristic approach to solving the problem are provided in Section 4.

4. Heuristic solution approaches

4.1. Heuristic for D-ICEP

As the computational results from Section 3.4.5 showed, for the objective functions that included the minimization of total evacuation time in the second stage, the S-ICEP seems to be much harder to solve for a commercial solver than with the sum of route times. As a consequence, commercial solvers are only able to solve relatively small instances in a reasonable amount of time. This paper aims to provide a first attempt to solve this problem in a fast and efficient way by using a structure-based heuristic to solve the D-ICEP and S-ICEP. This is motivated by the fact that, during an emergency situation, time is so valuable that it is crucial to obtain results quickly, and approximately optimal results are acceptable.

Bish (2011) provided a problem for bus evacuation with two heuristics that aim to solve the BEP efficiently. To solve D-ICEP efficiently, it was first attempted to solve the problem using the heuristics for the BEP. However, significant modifications were necessary. The compatibility between recovery resources and pick-up and drop-off nodes needs to be modeled, as well as the heterogeneity of the fleet in terms of capability and capacity. However, even with these modifications, these heuristics did not deliver solutions in any way close to the optimum, because the heuristic for the BEP takes advantage of the symmetry of the resource fleet, where all resources are considered identical and direct route swaps between resources are possible.

For the D-ICEP, the heterogeneity of the fleet and limited compatibility between nodes and resources make the problem a lot more complex and therefore require a different algorithm structure. While a structure that first generates an initial feasible solution first and then applies a local search to improve this solution can still be used, every step needs to include a feasibility check. In contrast to the BEP, the number of movements per resource does not play a large role; because of the fleet heterogeneity, a movement by one resource does not necessarily have the same impact as a movement by another resource. Hence, the expected evacuation time of a resource, along with its passenger capacity, have to be used as an allocation argument. This requires an inherently different structure of the local search heuristic. Algorithm 1, available in Appendix, is the first phase of the newly developed heuristic. It takes as inputs the fleet of resources, the evacuation locations, pick-up and drop-off nodes, number of evacuees, a travel distance matrix, and an upper bound to the evacuation time. It returns an evacuation plan that provides a good starting point for a local search that minimizes the total evacuation time.

Algorithm 1 uses a step-wise greedy structure that starts with the initial set-up at the beginning of an evacuation and greedily adds additional trips for each resource until all people are allocated to a trip to safety. It considers the initial time to availability of each resource in the initial route time of each resource. While there are still people left, the algorithm generates a potential next trip to the evacuation area and back for each resource, based on where there is demand, and which additional trip would result in the shortest expected total route time. The heuristic then selects the resource with the lowest expected total route time and makes the addition of this trip to its route permanent if its expected route time does not exceed the maximum evacuation time given as an input. Note that this is in accordance with the added upper time limit to D-ICEP considered in Section 3.3.4. The evacuees are then allocated based on the capacity of the resource. If the trip cannot be added without violating the maximum route time, then the while loop is interrupted and the route plan is returned without evacuating all the population. The provided route plan gives a feasible solution to the D-ICEP. The computational complexity can be identified as $O(mn(j + k))$ in the worst case, where m represents the number of evacuees, n represents the number of resources, j represents the number of pick-up nodes, and k represents the number of drop-off nodes. However, this worst case only applies if the capacity of a resource is 1. In the real world, resources can hold multiple passengers, so using a real-world data set reduces the time complexity significantly.

Because this first phase is a greedy algorithm, the solution is not guaranteed to be optimal. A local search heuristic was developed that tries to find better solutions by allocating remaining evacuees, reallocating evacuees to additional trips of other resources, and swapping entire trips between resources. Algorithm 2 can be reviewed in the Appendix and describes this second phase of the heuristic.

Algorithm 2 uses multiple strategies sequentially to improve the solution generated by Phase 1. It consists of a three-step structure that continues iterating through these steps until no improvement can be found. It considers the differences in speed profile, dock compatibility, and passenger capacity among resources to find solutions. The initialization starts with the route plan generated by Algorithm 1 and the maximum time limit. If remaining evacuation demand exists that could not be accommodated with the route plan from Algorithm 1, then Algorithm 2 at first tries to allocate the remaining evacuees. It does this by exploring, for every pick-up node with remaining evacuees, whether there is extra capacity on other resources that visit this pick-up node and whether these remaining evacuees can be reallocated while the plan still conforms to the maximum time limit. In its second and third steps, the algorithm tries to shorten the total evacuation time. The second step starts with the last trip of the resource with the longest evacuation time, here called the limiting resource, and tries to reallocate its passengers. For the chosen trip of the limiting resource, it first tries to allocate the passengers to excess capacity on existing trips of alternative resources. If this is not sufficient, it checks whether an alternative resource can perform an additional trip to the corresponding location and pick up some of the passengers, if this will not increase the current evacuation time. The algorithm continues iterating through the alternative resources until all passengers have been reallocated and the total evacuation time has been improved. If this is not the case after checking all alternative resources, the reallocation is canceled, and step 2 is repeated on another trip of the limiting resource.

If the second step does not lead to an improvement, the third step tries to swap trips between the limiting resource and alternative resources to reduce the overall evacuation time. For example, if resource 1 was serving route $B1 \rightarrow C1 \rightarrow B1 \rightarrow C2$, and resource 2 was serving $B2 \rightarrow C2 \rightarrow B1 \rightarrow C3$, a successful trip switch of the first trip would result in resource 1 performing $B2 \rightarrow C2 \rightarrow B1 \rightarrow C2$ and resource 2 performing $B1 \rightarrow C1 \rightarrow B1 \rightarrow C3$. Again starting with the last trip of the limiting resource, the algorithm stops once an improvement is found or once all trips of all alternative resources have been tested.

After every iteration, the list of resources gets updated. If, after an improvement, a different resource contains the most time-consuming route, this one will become the limiting resource. The algorithm keeps iterating until none of the steps lead to an improvement of the solution. The theoretical run time complexity of this algorithm will be assessed for each step at first. For step 1, the worst case run-time complexity is $O(mnj)$, where m is the number of evacuees, n is the number of resources, and j the number of pick-up nodes. For step 2, the approximate worst case run-time complexity is $O(m^2nk)$, where k is the number of pick-up nodes. For the worst case, step 3 provides an approximate computational complexity of $O(m^2nj)$. This results in a run time of $O(m^2nj)$, or $O(m^2nk)$ per iteration of the outer while loop, depending on whether set j or set k is larger. Similarly to phase 1 of the heuristic, note that the algorithm does not scale 1:1 with the number of evacuees, since resources generally have more than a passenger capacity of 1 and thus the number of trips is a fraction of the evacuee number. Furthermore, the fact that each step stops as soon as a valid improvement has been found makes the algorithm much faster in the average case. It is also not trivial to determine the complexity of the outer while loop, since it iterates until no more improvements can be found. Because the run time complexity is difficult to estimate theoretically, the next section provides numerical experiments on some test data sets.

4.2. Numerical experiments on the heuristic for D-ICEP

The two-phase heuristic introduced in the previous section was benchmarked against an implementation of the D-ICEP in the Pyomo interface with the Gurobi 9.0 commercial solver. Table 6 illustrates some key characteristics that describe the size of the test data sets. A variety of small to medium-sized data examples that reflect realistic scenarios for emergency evacuations of isolated areas were chosen for the benchmark. Table 7 describes the results of the computational tests. To account for the effect of the local search in the second phase of the heuristic, separate tests were conducted using only phase 1 and using both phase 1 and 2. All computational runs were completed on a MacBook Pro with a 2.6 GHz Dual-Core Intel Core i5 CPU. A run time limit of 3600 s was enforced. The settings were otherwise identical to the ones chosen in Section 3.4.5.

Table 7 shows that the presented D-ICEP heuristic was able to produce reasonable solutions for all but the smallest data sets, and, in several cases, also reached the optimal solution. Given that the presented heuristic is greedy, it could explore all areas of

Table 6
Key characteristics of the test data sets for the D-ICEP.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
Sets	Set size											
Resources	6	6	6	6	8	8	20	20	20	20	20	20
Resource loc.	1	1	3	3	3	3	3	3	3	3	3	3
Evac. locations	2	2	2	2	3	3	2	2	1	1	5	5
Pick-up points	4	2	3	3	4	4	4	3	2	2	8	8
Drop-off points	2	2	3	3	3	3	3	3	3	3	3	3
Round trips	6	6	8	8	8	8	20	20	20	20	20	20
Parameters	Setting											
Penalty							5,000					
Max route time							600					
Variable type	Quantity											
Continuous var.	303	157	446	446	783	785	4,829	3,628	1,124	1,124	9,639	9,639
Binary var.	312	156	450	450	800	800	4,880	3,660	1,220	1,220	9,760	9,760

Table 7
Result summary data experiments for the D-ICEP.

Data set	Implementation	Objective	Run-time	Iterations	Optimality gap	Run-time gap
D1	Gurobi	72	2.24 s	–	–	–
	Heuristic Phase 1	100	0.95 s	–	28.00%	−57.49%
	Heuristic Phase 1&2	90	2.34 s	1	20.00%	4.70%
D2	Gurobi	120	2.49 s	–	–	–
	Heuristic Phase 1	153.33	1.24 s	–	21.74%	−50.32%
	Heuristic Phase 1&2	153.33	3.92 s	1	21.74%	57.43%
D3	Gurobi	131.66	3.17 s	–	–	–
	Heuristic Phase 1	133.66	1.81 s	–	1.50%	−42.90%
	Heuristic Phase 1&2	131.66	3.18 s	2	–	0.32%
D4	Gurobi	81.67	4.38 s	–	–	–
	Heuristic Phase 1	81.67	0.63 s	–	–	−85.62%
	Heuristic Phase 1&2	81.67	0.89 s	1	–	−79.68%
D5	Gurobi	88.33	7.35 s	–	–	–
	Heuristic Phase 1	88.33	1.02 s	–	–	−86.12%
	Heuristic Phase 1&2	88.33	1.48 s	1	–	−79.86%
D6	Gurobi	93.6	5.32 s	–	–	–
	Heuristic Phase 1	93.6	1.86 s	–	–	−65.04%
	Heuristic Phase 1&2	93.6	2.07 s	1	–	−61.09%
D7	Gurobi	96	98.27 s	–	–	–
	Heuristic Phase 1	97.59	8.83 s	–	1.63%	−91.01%
	Heuristic Phase 1&2	96	17.85 s	2	–	−81.84%
D8	Gurobi	153.33	125.75 s	–	–	–
	Heuristic Phase 1	170.33	20.24 s	–	9.98%	−83.90%
	Heuristic Phase 1&2	164.24	17.85 s	2	6.64%	−70.74%
D9	Gurobi	77.2	50.53 s	–	–	–
	Heuristic Phase 1	77.2	2.96 s	–	–	−94.14%
	Heuristic Phase 1&2	77.2	4.61 s	2	–	−90.88%
D10	Gurobi	81.6	235.95 s	–	–	–
	Heuristic Phase 1	81.6	20.44 s	–	–	−91.34%
	Heuristic Phase 1&2	81.6	32.72 s	2	–	−86.13%
D11	Gurobi	252.24[a]	3600 s[a]	–	(lb 245.93) 2.64%	–
	Heuristic Phase 1	275.6	113.45 s	–	10.7%[b]	−96.85%[c]
	Heuristic Phase 1&2	275.6	209.95 s	1	10.7%[b]	−94.17%[c]
D12	Gurobi	276.24[a]	3600 s[a]	–	(lb 262.99) 4.80%	–
	Heuristic Phase 1	300.66	134.83 s	–	12.56%[b]	−96.25%[c]
	Heuristic Phase 1&2	282.56	249.42	3	6.95%[b]	−93.07%[c]

[a]Results were aborted after 3600 s; the best available solution, optimality gap and lower bound are displayed.
[b]Optimality gap estimated based on lower bound provided by Gurobi 9.0.
[c]Run-time reduction compared to run time limit of 3600 s.

the solution space, and was not guaranteed to find the global optimum. In cases D4, D5, D6, D9, and D10, the first phase was able to reach the optimal solution. In cases D3 and D7, the first phase was not able to generate an optimal solution by itself, and the optimal solution could be found when the local search algorithm from phase 2 was also used. In cases D1, D8, D11, and D12, neither using only phase 1, nor both phase 1 and 2 was sufficient to reach the optimal solution, but phase 2 improved the solution quite significantly and reduced the optimality gap for D1, D8, and D12. Using only the first phase of the heuristic led to considerably faster run times, and, in many cases, better solutions. With regards to algorithm run time, observe that for most smaller problems (D1–D3), the heuristic did not lead to significant improvements in run time, and the desire for a good solution quality makes solving problems of this size with a commercial solver more attractive. However, the run time of Gurobi 9.0 increased significantly over a growing problem size, as the larger test cases showed. While the heuristic run time increased, too, its growth rate was much smaller in practice than the worst case theoretical run time from the previous section hinted. Note also that the experiment results show that the local search procedure only ran for a few iterations until it could not find a better solution, which demonstrated a problem from the previous question that the number of iterations of phase 2 of the heuristic was non-trivial to estimate. This showed that, for larger problem sizes, while the presented heuristic is not guaranteed to find a global optimum, it produces a solution much more quickly than a commercial solver does.

Therefore, when solving a larger instance of the D-ICEP, the decision whether to use a commercial solver or the presented heuristic should be based on whether the solution quality or the run time is more important. In emergency situations having a good solution quickly is often preferable to waiting for a better solution. In the experiments, none of the steps in either phase 1 or phase 2 of the heuristic were parallelized in execution, but rather executed sequentially. Hence, there is still potential for improvement when using multi-core processors that can execute process steps in parallel. Nevertheless, it is not possible to establish a reliable bound on how close the heuristic will get to the optimal solution, because that is dependent on the exact problem instance and

could only be approximated by conducting additional experiments with various data sets. When it tackles bigger problem sizes, the heuristic's run time increases, but it is able to handle larger problem sizes in reasonable time with the trade-off that the global optimal solution might not be found. Further expanding the heuristic to explore additional parts of the solution space might lead to a better solution quality, but will also increase the run time further, thus resulting in a trade-off. An alternative approach to finding the global optimum in approximation without increasing run time could be to use a metaheuristic.

When there is no emergency, and the model is being used for planning, there is less need for speed, but larger instances can still generate problems for commercial solvers, as the studies in Section 3.4.5 show. The following section therefore explores how to use the D-ICEP heuristic presented above to solve the S-ICEP.

4.3. Heuristic for the S-ICEP

The analysis in Section 3.4.5 showed that the primary objective function *Bal_1* is most difficult to solve for a commercial solver. Since it is also the primary objective function of the S-ICEP, this section focuses on solving the S-ICEP with this objective function efficiently. Using the heuristic developed for the D-ICEP, a framework that can find a solution to the S-ICEP is introduced. It consists of a greedy heuristic search framework that starts with an empty resource set and adds a resource to the fleet on every iteration, depending on whether adding it improves the total evacuation plan cost. The algorithm terminates if no additional resource improves the solution or if all available resources have been added to the resource fleet. Algorithm 3 in the Appendix describes this algorithm.

Algorithm 3 is structured similarly to the D-ICEP heuristic in that it greedily selects a resource to be part of the solution set if that resource improves the solution in expectation. However, to reduce algorithm run time, the algorithm always adds a resource into the set if it provides an improvement and does not consider whether another resource would have provided a larger gain. Alternatively, a more involved approach could be chosen in which the impact of adding every possible resource is tested before any are selected. But this would greatly increase the run time, because there is no simple way to determine which resource will provide the biggest improvement. Taking into account the results of the experiments on the D-ICEP heuristic showing that the local search heuristic does not iterate much until it cannot find further improvements, a theoretical run time from the D-ICEP of $O(m^2nj)$ is obtained. If the algorithm were to test the addition of each potential resource every iteration, this would result in a theoretical run time of $O(n^3m^2tj)$ for the entire algorithm, where n is the number of resources, m is the number of evacuees, t is the number of scenarios and j is the number of pick-up nodes. Therefore, the solution described in Algorithm 3 was chosen, which returns a worst case theoretical run time of $O(n^2m^2tj)$, if both phases of the D-ICEP heuristic are run. If only the first phase is run, the worst case run time is $O(n^2m(j+k))$, where k is the number of drop-off nodes. While this approach reduces the share of the solution space that is explored, it allows us to find a solution more quickly.

The run time can also be reduced by providing the algorithm with an initial resource set instead of starting with an empty set. The risk of missing the global optimum through this approach is low if parameter Q is not too large, since a small resource set generally leads to longer evacuation times. The rule for determining the initial resource set is presented in Algorithm 4, available in Appendix.

Algorithm 4 selects the initial resource set on the basis of (1) whether, for every scenario, every pick-up node that has evacuation demand can be served and (2) whether it is possible to cover at least a certain percentage of evacuation demand at each pick-up node if it is visited by only one trip of each resource. The worst case theoretical run time of this algorithm is $O(jn)$ or $O(tj)$, where j is the set of evacuation pick-up points, n is the set of potential resources, and t is the set of scenarios, depending on whether tj or jn are larger. A warm start can be considered, where the problem is provided with an initial resource set that is capable of evacuating all scenarios, instead of starting from an empty resource set. In the following section, the variants of the S-ICEP heuristic are discussed regarding their run time and solution quality, on the basis of four test instances, similar to the experimental results presented for the D-ICEP in the previous section.

4.4. Numerical experiments on the heuristic for the S-ICEP

This section describes tests of the developed S-ICEP heuristic in comparison to those of the Gurobi 9.0 commercial solver. Four test data sets were used, which are presented in Table 8, to illustrate the performance of the S-ICEP heuristic in comparison to Gurobi 9.0.

Because the S-ICEP heuristic makes use of the D-ICEP heuristic, the structural challenges that the D-ICEP faces also apply to this algorithm. However, testing the algorithm showed whether to use both phase 1 and 2 of the D-ICEP. It also showed the effect of a warm start. Table 9 provides the results of the computational tests with the data sets in Table 8 for Gurobi 9.0, the greedy S-ICEP heuristic discussed in Section 4.3 using only the first stage, or both the first and second stages of the D-ICEP heuristic, and the S-ICEP heuristic, starting with an initial route set defined by the warm start rule (indicated by "+ WS") introduced in Algorithm 4 with parameter Q set to 20 percent.

Table 9 shows that in contrast to the D-ICEP, no tested configuration allowed the S-ICEP heuristic to reach the global optimum of the *Bal_1* objective function of the S-ICEP. This is caused by the fact that the D-ICEP does not guarantee to find the global optimum, and the assumptions generated for the S-ICEP heuristic on resource selection also further simplify the problem. In some cases, using both phase 1 and phase 2 of the D-ICEP reduced the optimality gap. The run time of the algorithm, especially for larger problems, could also be significantly reduced through using the warm start for the initial resource set without sacrificing the solution quality of the S-ICEP heuristic. It is therefore recommended that the warm start feature be used if larger problems are investigated. In future research, additional numerical experiments can be conducted to investigate ideal parameter settings for parameter Q. Given the

Table 8
Test data sets for the S-ICEP.

Sets	S1	S2	S3	S4
	Set size			
Scenarios	2	3	4	6
Potential resources	6	6	8	20
Initial storage locations	1	2	3	6
Evacuation locations	3	4	5	8
Evacuation pick-up points	5	6	8	15
Safe drop-off points	2	3	3	6
Round trips (only for commercial solver)	6	8	8	20
Parameters	Setting			
Penalty	5,000	5,000	5,000	5,000
Evacuation time limit (min)	120	240	500	600
Variable type	Quantity (for commercial solver)			
Continuous variables	832	2,718	6,640	109,668
Binary variables	858	3,198	8,324	178,740

Table 9
Result summary data experiments for the S-ICEP.

Data set	Implementation	Objective	Run-time	Optimality gap	Run-time gap
S1	Gurobi	102.21	10.93 s	–	–
	S-ICEP (incl. D-ICEP Ph. 1)	148.77	7.56 s	31.30%	−30.83%
	S-ICEP (incl. D-ICEP Ph. 1&2)	148.77	10.7 s	31.30%	2.10%
	S-ICEP (incl. D-ICEP Ph. 1 + WS)	148.77	4.91 s	31.30%	−55.08%
	S-ICEP (incl. D-ICEP Ph. 1&2 + WS)	148.77	8.61 s	31.30%	−21.23%
S2	Gurobi	119.65	3600 s	(bb 119.6) 0.04%	–
	S-ICEP (incl. D-ICEP Ph. 1)	155.6	11.39 s	23.13%	−99.68%
	S-ICEP (incl. D-ICEP Ph. 1&2)	142.04	30.3 s	15.80%	−99.16%
	S-ICEP (incl. D-ICEP Ph. 1 + WS)	155.6	5.63 s	23.14%	−99.78%
	S-ICEP (incl. D-ICEP Ph. 1&2 + WS)	142.04	27.91 s	15.80%	−99.22%
S3	Gurobi	120.19	144.83 s	–	–
	S-ICEP (incl. D-ICEP Ph. 1)	162.6	51.65 s	26.08%	−64.34%
	S-ICEP (incl. D-ICEP Ph. 1&2)	162.32	151.39 s	25.95%	4.53%
	S-ICEP (incl. D-ICEP Ph. 1 + WS)	162.6	13.35 s	26.08%	−90.78%
	S-ICEP (incl. D-ICEP Ph. 1&2 + WS)	162.32	31.07 s	25.95%	−78.55%
S4	Gurobi	143.30[a]	3600 s[a]	(bb 109.77) 23.4%	–
	S-ICEP (incl. D-ICEP Ph. 1)	169.02	1937.11 s	35.06%[b]	−46.19%[c]
	S-ICEP (incl. D-ICEP Ph. 1&2)	172.48	2954.66 s	36.36%[b]	−17.93%[c]
	S-ICEP (incl. D-ICEP Ph. 1 + WS)	169.64	576.87 s	35.29%[b]	−83.97%[c]
	S-ICEP (incl. D-ICEP Ph. 1&2 + WS)	167.42	1011.9 s	34.43%[b]	−71.89%[c]

[a] Results were aborted after 3600 s; the best available solution, optimality gap and best bound are displayed.
[b] Optimality gap estimated based on lower bound provided by Gurobi 9.0.
[c] Run-time reduction compared to run time limit of 3600 s.

experiments presented in Table 9, it is evident that the S-ICEP heuristic is able to reduce the run time to find a reasonable solution for larger problems in comparison to an implementation of a commercial solver. However, the solution quality suffers significantly and to a much higher degree than for the D-ICEP heuristic. The reason is that the gaps in the second stage add up for each scenario and that the resource selection in the first stage of the heuristic is not reliable in finding the best resource set.

Despite the possibility of reducing the algorithm's run time through these tweaks, two main caveats of the logic-based heuristic for the S-ICEP remain. First, the algorithm run time of the S-ICEP heuristic still increases significantly with the problem size. While the increase does not happen at the same rate as that of the commercial solver, the solution quality is sacrificed significantly because the optimality of the solution cannot be guaranteed. Second, the complexity of the ICEP makes it difficult to efficiently explore the solution space if a structure-based approach is used. It is possible to add additional improvement checks to phase 2 of the D-ICEP heuristic, but additional features and layers increase the run-time complexity and thus also reduce the usability of the algorithm for larger problem sizes. The S-ICEP heuristic should therefore only be used if a solution needs to be obtained as quickly as possible. For planning purposes, the commercial solver is thus preferred, until an alternative solution method is available.

5. Conclusions

This paper introduced the ICEP and its variants D-ICEP and S-ICEP, which aim to improve emergency planning by optimizing routing and fleet selections for the evacuation of isolated communities. The special considerations of D-ICEP on heterogeneous

fleets, limited compatibility between nodes and resources, and alternative ways of evacuation make the ICEP models more difficult to solve efficiently than previous models, but they enable emergency planners to develop an evacuation plan that is applicable to remote areas. This is the first routing model developed for planning the evacuation of isolated communities with a coordinated set of resources and therefore delivers an important contribution for research and practice on this topic. The S-ICEP allows scenario-based planning by optimizing the evacuation resource set over multiple disaster scenarios that differ in evacuee numbers, locations, and weather. This makes the model framework compatible with common evacuation planning practices. Alternative objective functions of varying risk levels were presented and explored by conducting numerical experiments. In addition, guidance on the use of both models and its parameter settings was provided. Moreover, heuristic solution approaches were presented to solve the problems quickly. A two-phase structure-based greedy search heuristic was presented for the D-ICEP. Computational experiments showed that this heuristic is able to significantly reduce the run time of the algorithm and it found the optimal solution in some cases or got reasonably close, but this was not the case for all test data sets. Building on the D-ICEP heuristic, a greedy search heuristic was developed to select evacuation resources for the S-ICEP. Experiments showed that this heuristic also reduced the algorithm run time significantly in comparison to a commercial solver, but it did not reach the global optimum in any test case and showed large optimality gaps.

Future research could focus on the following model extensions:

- Relaxing the constraints that allow resources to visit only a single evacuation pick-up point per trip.
- A manual prioritization feature that allows the modeler to specify a specific region to be evacuated first. This can be helpful if a certain region is closer to the danger zone and needs to be prioritized during evacuation.
- Expanding the model to include the transportation of evacuees to the pick-up nodes.

There are several possible strategies for modeling the movement of evacuees to pick-up locations. The D-ICEP could be integrated with a flow network that either reconfigures the demand at each location, similar to Wang and Wang (2019), or simulates an arrival rate. Or, rolling horizon implementations of D-ICEP could be used that model the arrival of evacuees through sequential updates, resolving the remaining, not yet executed, part of the solution, every time new information is obtained. Robust optimization could be another approach.

As mentioned in the Introduction section (Section 1), a real-world case study can be helpful to investigate the value of the model for practitioners and derive managerial insights in detail, and to learn more about the evacuation of isolated communities and explore how this model can best be applied. As mentioned in the discussion about scenario generation (Section 3.3.2), it can be challenging to obtain reliable data to solve the S-ICEP problem using existing data sets. Thus, subject matter experts should be included in the data design process, and close collaboration with first responders is necessary to make sure that the data assumptions provided to the model as inputs are realistic. Krutein et al. (2022) provide such a case study for an isolated island. Applications to other types of isolated communities can be promising for more insights. Future research could further focus on additional algorithmic solutions. One option is to expand the presented heuristic algorithm to search a larger share of the solution space, while balancing out the trade-off with computational complexity. Alternative solution methods on how to solve the ICEP models to a more reliable solution quality could consider metaheuristic frameworks or decomposition methods.

CRediT authorship contribution statement

Klaas Fiete Krutein: Conceptualization, Project administration, Methodology development, Formal analysis, Data curation, Software programming, Writing – original draft, Visualization. **Anne Goodchild:** Funding acquisition, Supervision, Resources provision, Writing – review & editing.

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.

Acknowledgments

Appendix. Algorithms

Algorithm 1: D-ICEP Heuristic Phase 1: Initial Feasible Solution Generation

---

    Result: A feasible evacuation route plan

1 initialize all resources n, evacuees m as the remaining evacuees, pick-up nodes j, drop-off nodes k;

2 set a maximum time for the route plan;

3 **while** <u>remaining evacuees > 0</u> **do**

4   **for** <u>n in resources</u> **do**

5   **for** <u>j in pick-up nodes</u> **do**

6   Calculate the distance to the current location of n;

7   **end**

8   Select the closest pick-up node to n as the next potential pick-up node;

9   **for** <u>k in drop-off nodes</u> **do**

10   Calculate the distance to the next potential pick-up node of n;

11   **end**

12   Select the closest drop-off node to the next potential pick-up node of n as the next potential drop-off node;

13   Expected route time $[n]$:= current route time$[n]$ + time to next potential pick-up node + load time + time to next potential drop-off node + unload time

14   **end**

15   Select the resource a with the lowest expected route time;

16   **if** <u>expected route time $[a]$ ≤ max route time</u> **then**

17   Next pick-up node $[a]$:= Next potential pick-up node$[a]$;

18   Next drop-off node $[a]$:= Next potential drop-off node$[a]$;

19   **if** <u>remaining demand at next pick-up node > capacity of resource a</u> **then**

20   load evacuees according to max capacity;

21   **else**

22   load evacuees according to remaining demand;

23   **end**

24   Update current route time $[a]$;

25   **else**

26   break and return incomplete route plan;

27   **end**

28 **end**

Algorithm 2: D-ICEP Heuristic Phase 2: Improvement through Local Search

---

    Result: An improved feasible evacuation route plan

1 Input a feasible route plan, max route time;

2 Initialize resources n and pick-up nodes j and drop-off nodes k accordingly;

3 Extra demand added := re-allocation added := swap added := True;

4 **while** Extra demand added = True OR re-allocation added = True OR swap added = True **do**

5   **1. Try allocating extra demand**;

6   Extra demand added := False;

7   **for** j in pick-up nodes with remaining demand **do**

8   Sort resources by current route time in ascending order; $n := 1$;

9   **while** extra demand left AND not all resources checked **do**

10   **if** any trip of resource n serves j and has excess capacity **then**

11   Allocate extra demand until capacity of n is exhausted; Extra demand added := True;

12   **end**

13             $n := n + 1$;

14   **end**

15   **end**

16   **2. Try re-allocating passengers from the longest route**;

17   Select the resource l with the longest evacuation time as the limiting resource;

18   Re-allocation added := False; $n := 1$; $z :=$ no. of trips on resource l;

19   **while** Re-allocation added = False AND $z \geq$ no. of trips on resource l **do**

20   **while** Trip z of l has remaining passengers AND n $\leq$ no. of alternative resources **do**

21   **if** any trip of alternative resource n serves j and has excess capacity **then**

22   Re-allocate passengers; **if** no more passengers **then** break loop;

23   **end**

24   **if** Trip z of l has remaining passengers AND an additional trip of resource n serving pick-up node j and the closest drop-off node k can be added without exceeding the current evacuation time **then**

25   Re-allocate passengers; **if** no more passengers **then** break loop;

26   **end**

27             $n := n + 1$;

28   **end**

29   **if** trip z of resource l has no more passengers AND the total route time through this change < current route time **then**

30   Re-allocation added := True; make re-allocation permanent; break loop;

31   **else**

32   Reverse the re-allocation of trip z;

33             $z := z - 1$;

34   **end**

35   **end**

36   **3. Try swapping routes between resources to decrease evacuation time**;

37   **if** Re-allocation added = False **then**

38   Swap added := False; $n := 1$; $z :=$ no. of trips on resource l;

39   **while** Swap added = False AND $z \geq$ no. of trips on resource l **do**

40   **while** $n \leq$ no. of alternative resources **do**

41   **if** any trip of alternative resource n can be swapped with trip z of l **then**

42   Perform the swap; Swap added := True; break loop;

43   **else**

44                  $n := n + 1$;

45   **end**

46              $z := z - 1$;

47   **end**

48   **end**

49   **end**

50 **end**

Algorithm 3: S-ICEP Heuristic: Optimal resource fleet

   Result: A cost- and time-efficient evacuation plan

1   initialize the best cost as the penalty cost inflicted by not evacuating any person;
2   initialize the current resource fleet as an empty set;
3   initialize $n := 0$;
4   **while** <u>current cost < best cost AND not all resources are in the resource fleet</u> **do**
5   best cost = current cost;
6   sort the list of candidate resources by maximum capacity, time to availability and maximum speed;
7   Add the nth entry of the list of candidate resources to the set of resources;
8   **for** <u>t in scenarios</u> **do**
9   Run D-ICEP Heuristic Phase 1 with set of resources;
10   Run D-ICEP Heuristic Phase 2 with set of resources;
11   **end**
12   proposed cost = total cost of the S-ICEP evacuation plan considering scenario probabilities and cost parameters;
13   **if** <u>proposed cost < current cost</u> **then**
14   current cost = proposed cost;
15   delete the nth resource from the list of candidate resources;
16   **else**
17   delete the nth resource from the resource set again;
18        $n += 1$;
19   **if** <u>n equals the length of the candidate resource set</u> **then**
20   break the loop;
21   **end**
22   **end**
23 **end**

Algorithm 4: S-ICEP Heuristic: Initial resource set selector

   Result: A set of resources usable as a warm start to the S-ICEP heuristic

1   input the set of potential resources and the set of evacuation pick-up points including the evacuation demand for each scenario;
2   provide a minimum percentage of evacuation demand that needs to be covered by resources if only one trip to each evacuation location by every resource is completed and initialize as Q;
3   initialize the initial resource set as an empty set;
4   initialize the set of total pick-up points as an empty set;
5   **for** <u>t in scenarios</u> **do**
6   **for** <u>j in evacuation pick-up points</u> **do**
7   **if** <u>j has evacuation demand and is not yet in the set of total pick-up points</u> **then**
8   Add j to the set of total pick-up points
9   **end**
10   **end**
11 **end**
12 **for** <u>i in total evacuation pick-up points</u> **do**
13   Order the list of potential resources compatible with i by maximum capacity, time to availability and maximum speed;
14      $n := 0$ **while** <u>total evacuation capacity at node i in scenario j < Q(evacuation demand at i)</u> **do**
15   **if** <u>nth resource not in set of initial resources yet</u> **then**
16   add the nth resource to the initial resource set;
17   **else**
18          $n := n + 1$
19   **end**
20   **end**
21 **end**

References

Abdelgawad, H., Abdulhai, B., 2010. Managing large-scale multimodal emergency evacuations. J. Transp. Saf. Secur. 2 (2), 122–151.

An, S., Cui, N., Li, X., Ouyang, Y., 2013. Location planning for transit-based evacuation under the risk of service disruptions. Transp. Res. B 54, 1–16.

Australian Broadcasting Corporation, 2020. Mallacoota evacuations begin as thousands trapped by bushfires are transported to navy ships. ABC News, 2 Jan.

Baou, E., Koutras, V.P., Zeimpekis, V., Minis, I., 2018. Emergency evacuation planning in natural disasters under diverse population and fleet characteristics. J. Humanit. Logist. Supply Chain Manage. 8 (4), 447–476.

Bayram, V., 2016. Optimization models for large scale network evacuation planning and management: A literature review. Surv. Oper. Res. Manag. Sci. 21 (2), 63–84.

Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Wolfler Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Comput. Oper. Res. 38 (6), 931–941.

Bellman, R., 1966. Dynamic programming. Science 153 (3731), 34–37.

Bish, D.R., 2011. Planning for a bus-based evacuation. OR Spectrum 33 (3), 629–654.

Brandão, J.C.S., Mercer, A., 1998. The multi-trip vehicle routing problem. J. Oper. Res. Soc. 49 (8), 799–805.

Britten, L., 2019. How do you evacuate all of Bowen Island? Municipality seeks plan to find out. CBC News, 31 Mar.

Burkard, R.E., Dlaska, K., Klinz, B., 1993. The quickest flow problem. ZOR, Methods Models Oper. Res. 37, 31–58.

Cattaruzza, D., Absi, N., Feillet, D., 2016. The multi-trip vehicle routing problem with time windows and release dates. Transp. Sci. 50 (2), 676–693.

Coffey, H., 2019. Greece wildfires: Hundreds of tourists evacuated from hotels and beaches. The Independent, 27 Aug.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54 (3), 573–586.

Crevier, B., Cordeau, J.F., Laporte, G., 2007. The multi-depot vehicle routing problem with inter-depot routes. European J. Oper. Res. 176 (2), 756–773.

Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. Manage. Sci. 6 (1), 80–91.

Deane, K., Coto, D., 2021. 'Huge' explosion rocks St. Vincent as volcano keeps erupting. Associated Press, 12 Apr.

Dikas, G., Minis, I., 2016. Solving the bus evacuation problem and its variants. Comput. Oper. Res. 70, 75–86.

Flood, M.M., 1956. The travelling-salesman problem. Oper. Res. 4 (1), 61–75.

Ford, L.R., Fulkerson, D.R., 1958. Constructing maximal dynamic flows from static flows. Oper. Res. 6 (3), 419–433.

Goerigk, M., Deghdak, K., Heßler, P., 2014a. A comprehensive evacuation planning model and genetic solution algorithm. Transp. Res. E 71, 82–97.

Goerigk, M., Deghdak, K., T'Kindt, V., 2015. A two-stage robustness approach to evacuation planning with buses. Transp. Res. B 78, 66–82.

Goerigk, M., Grün, B., 2014. A robust bus evacuation model with delayed scenario information. OR Spectrum 36 (4), 923–948.

Goerigk, M., Grün, B., Heßler, P., 2013. Branch and bound algorithms for the bus evacuation problem. Comput. Oper. Res. 40 (12), 3010–3020.

Goerigk, M., Grün, B., Heßler, P., 2014b. Combining bus evacuation with location decisions: A branch-and-price approach. Transp. Res. Procedia 2, 783–791.

Gorissen, B.L., Yanikoglu, I., den Hertog, D., 2015. A practical guide to robust optimization. Omega 53, 124–137.

IPCC, 2012. Managing the risks of extreme events and disasters to advance climate change adaptation a special report of working groups I and II of the intergovernmental panel on climate change [C.B. Field, V. Barros, T.F. Stocker, D. Qin, D.J. Dokken, K.L. Ebi, M.D.]. Intergovernmental Panel on Climate Change, p. 582.

Karabuk, S., Manzour, H., 2019. A multi-stage stochastic program for evacuation management under tornado track uncertainty. Transp. Res. E 124 (February), 128–151.

Krutein, K.F., McGowan, J., Goodchild, A., 2022. Evacuating isolated Islands with marine resources: A Bowen Island case study. Int. J. Disaster Risk Reduct. 72 (102865).

Kulshrestha, A., Lou, Y., Yin, Y., 2014. Pick-up locations and bus allocation for transit-based evacuation planning with demand uncertainty. J. Adv. Transp. 48, 721–733.

Laporte, G., 2009. Fifty years of vehicle routing. Transp. Sci. 43 (4), 408–416.

Laporte, G., Louveaux, F., Mercure, H., 1992. Vehicle routing problem with stochastic travel times. Transp. Sci. 26 (3), 161–170.

Lim, G.J., Rungta, M., Baharnemati, M.R., 2015. Reliability analysis of evacuation routes under capacity uncertainty of road links. IIE Trans. 47 (1), 50–63.

Lu, Q., George, B., Shekhar, S., 2005. Capacity constrained routing algorithms for evacuation planning: A summary of results. In: SSTD'05: Proceedings of the 9th International Conference on Advances in Spatial and Temporal Databases, Vol. 5644. pp. 291–307.

Moussa, M., Rising, D., 2022. Three of Tonga's smaller islands badly damaged by tsunami. Associated Press, 19 Jan.

Nadeau, B.L., 2021. Residents evacuated from Italian island Vulcano over carbon dioxide levels. p. 1, CNN, 21 Nov.

Pereira, V.C., Bish, D.R., 2015. Scheduling and routing for a bus-based evacuation with a constant evacuee arrival rate. Transp. Sci. 49 (4), 853–867.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. European J. Oper. Res. 225 (1), 1–11.

Pillac, V., Van Hentenryck, P., Even, C., 2016. A conflict-based path-generation heuristic for evacuation planning. Transp. Res. B 83, 136–150.

Renne, J.L., Sanchez, T.W., Litman, T., 2011. Carless and special needs evacuation planning: A literature review. J. Plan. Lit. 26 (4), 420–431.

Romero, D., 2019. Over 1,000 Bahamas residents evacuated after Dorian as food, supplies are brought in. NBC News, 7 Sep.

Sayyady, F., Eksioglu, S.D., 2010. Optimizing the use of public transit system during no-notice evacuation of urban areas. Comput. Ind. Eng. 59 (4), 488–495.

Schrage, L., 1981. Formulation and structure of more complex/realistic routing and scheduling problems. Networks 11 (2), 229–232.

Shapiro, A., 2008. Stochastic programming approach to optimization under uncertainty. Math. Program. 112, 183–220.

Sherali, H.D., 1982. Equivalent weights for lexicographic multi-objective programs: Characterizations and computations. European J. Oper. Res. 11 (4), 367–379.

Song, R., He, S.W., Zhang, L., 2009. Optimum transit operations during the emergency evacuations. J. Transp. Syst. Eng. Inf. Technol. 9 (6), 154–160.

Southworth, F., 1991. Regional Evacuation Modeling: A State-of-the-Art Review. Oak Ridge National Laboratory.

Soyster, A.L., 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. Oper. Res. 21 (5), 1154–1157.

StreetLight Data Inc., 2020. National map of communities with limited evacuation routes.

Thompson, R.R., Garfin, D.R., Silver, R.C., 2017. Evacuation from natural disasters: A systematic review of the literature. Risk Anal. 37 (4), 812–839.

Tüydeş, H., 2005. Network Traffic Management under Disaster Conditions (Ph.D. thesis). Northwestern University.

Wang, Y., Wang, J., 2019. Integrated reconfiguration of both supply and demand for evacuation planning. Transp. Res. 130 (August), 82–94.

Wu, L., Yang, D., Wang, S., Yuan, Y., 2020. Evacuating offshore working barges from a land reclamation site in storm emergencies. Transp. Res. 137 (January), 101902.

Zheng, H., 2014. Optimization of bus routing strategies for evacuation. J. Adv. Transp. 48, 734–749.