Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

A META-HEURISTIC SOLUTION APPROACH TO ISOLATED EVACUATION PROBLEMS

Klaas Fiete Krutein
Linda Ng Boyle

Department of Industrial & Systems Engineering
University of Washington
Box 352650
Seattle, WA 98195, USA

Anne Goodchild

Department of Civil & Environmental Engineering
University of Washington
201 More Hall, Box 352700
Seattle, WA 98195, USA

ABSTRACT

This paper provides an approximation method for the optimization of isolated evacuation operations, modeled through the recently introduced Isolated Community Evacuation Problem (ICEP). This routing model optimizes the planning for evacuations of isolated areas, such as islands, mountain valleys, or locations cut off through hostile military action or other hazards that are not accessible by road and require evacuation by a coordinated set of special equipment. Due to its routing structure, the ICEP is NP-complete and does not scale well. The urgent need for decisions during emergencies requires evacuation models to be solved quickly. Therefore, this paper investigates solving this problem using a Biased Random-Key Genetic Algorithm. The paper presents a new decoder specific to the ICEP, that allows to translate in between an instance of the S-ICEP and the BRKGA. This method approximates the global optimum and is suitable for parallel processing. The method is validated through computational experiments.

1 INTRODUCTION

This paper presents an approximate solution approach for isolated evacuation operations using the Isolated Community Evacuation Problem (ICEP) (Krutein and Goodchild 2022). The ICEP is a routing problem that models the evacuation of an isolated area using a fleet of coordinated resources when road-based evacuation is not possible. Some examples of recent events where coordinated resource routing for evacuation was necessary that demonstrate the relevance of this modeling approach, are the bush fires in Mallacoota, Australia in 2020 (ABC News 2020), and the evacuation of Samos Island, Greece in 2019, (Coffey 2019), which both required a coordinated marine and air evacuation from beaches to evacuate populations that were encircled by fires. This led to increased interest in building comprehensive evacuation plans for isolated areas (Britten 2019). The problem generates an optimal route plan for the resource fleet that minimizes the total evacuation time of the area. It is therefore a combination of a network flow problem and a routing problem. Since the ICEP contains a routing component it requires some binary variables and is formulated as a mixed-integer program (MIP). Commercial solvers that solve the problem for an exact optimal solution therefore need to use solution methods appropriate for integer programming, such as Lagrangian relaxations, plane cutting and branch-and-bound/branch-and-cut methods to find the optimal solution. For large problem instances, it can take a significant amount of time to find the optimal solution since the trees generated by the branching methods grow exponentially in size. An additional challenge is posed by using an objective function that minimizes the total evacuation time instead of the sum of route times, which is commonly used in vehicle routing problems (VRP), since the total evacuation time is defined as the duration of the longest route generated among the resources. Experiments have shown that solution discrimination is significantly more challenging for a commercial solver with such an objective function compared to minimizing the sum of route times (Krutein and Goodchild 2022). Since the problem aims to find a solution to respond to an emergency, it can be crucial to find a high quality solution quickly, and a near-optimal solution will suffice. A constructive heuristic that combines a greedy route plan generation with a local search has already been presented (Krutein and Goodchild 2022), which produces results quickly. However, this heuristic does not reliably produce solutions close to the optimal solution, since it can get stuck at local minima. This is particularly the case for the stochastic version of the ICEP, denoted S-ICEP, which takes uncertainty over evacuee numbers and event location into account. The reason why the S-ICEP is of particular relevance to evacuation planning is that during evacuation planning, it is often not known at all where exactly the source of a disaster is located, what populations are affected and where they should be evacuated from. For that reason using a scenario-based two-stage stochastic approach with weighted probabilities allows planners to identify the evacuation resources that perform best across a variety of potential disaster scenarios. On the instances tested with the simpler heuristic for S-ICEP, this led to optimality gaps of up to 36.36% (Krutein and Goodchild 2022), which is not practical for evacuation planning. While some in-depth analysis on the application challenges and data requirements for this modeling framework have already been obtained through a detailed real-world case study (Krutein et al. 2022), solving the problem quickly is still a challenge. The proposed approximation method for the S-ICEP leverages a meta-heuristic framework that uses a Multi-Parent Biased Random-Key Genetic Algorithm (MP-BRKGA) (Andrade et al. 2021). The paper introduces a S-ICEP specific decoder method that can be used to translate outputs from the MP-BRKGA into a unique solution to the S-ICEP. The method is validated by computational experiments. This paper provides the following contributions:

- A decoder algorithm that translates standardized MP-BRKGA outputs into a solution to the S-ICEP, enabling the approximation of the optimal solution to the S-ICEP.
- Experimental results that demonstrate better performance of the proposed method compared to commercial problem solvers.

2 THE ISOLATED COMMUNITY EVACUATION PROBLEM

The ICEP (Krutein and Goodchild 2022) aims to minimize the total evacuation time of an isolated area. The model includes different sets of nodes that capture resource staging locations, evacuation nodes, and safe nodes. The use of multiple trips of heterogeneous resources expands the network with each additional resource i and each additional trip k, which greatly increases the model complexity. To make the model more suitable for planning purposes, a two-stage stochastic version of the ICEP, denoted S-ICEP (Krutein and Goodchild 2022) was developed. The formulation of the S-ICEP is provided in Section 2.1 with notations in Table 1. This model seeks to find the optimal set of evacuation resources while also accounting for the parameters related to uncertainty in the emergency. A detailed explanation is provided by Krutein and Goodchild (2022).

2.1 Model Formulation

$$\begin{aligned} \min \quad & \frac{\sum_{i\in I} cf_i(z_i)}{\sum_{i\in I}(cf_i + cv_i(T))} + \mathbb{E}[C(z,\xi)] \\ s.t. \quad & z_i \in \{0,1\} && \forall i \in I \end{aligned}$$

where

$$\begin{aligned} C(z,\xi) := \min \quad & r + \frac{\sum_{i\in I} cv_i(s_i)}{\sum_{i\in I}(cf_i + cv_i(T))} + P\sum_{a\in A} n_a \\ s.t. \quad & r \geq s_i && \forall i \in I \end{aligned}$$

Table 1: Notation Key for S-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^i_{hb}	$\frac{distance(h \rightarrow b)}{\text{empty travel speed of resource } i}$: cost of arc ζ^{1i}_{hb}
$h \in H$	initial resource staging locations	t^i_{bc}	$\frac{distance(b \rightarrow c)}{\text{loaded travel speed of resource } i}$: cost of arc γ^{ki}_{bc}
$\xi \in \Xi$	evacuation scenarios	t^i_{cb}	$\frac{distance(c \rightarrow b)}{\text{empty travel speed of resource } i}$: cost of arc δ^{ki}_{cb}, only $k = 1, ..., K-1$
		cf_i	fixed cost parameter of resource i
		cv_i	variable cost parameter of resource i
		T	upper time limit for r
		P	penalty cost for non-evacuated person

Arcs	Arc Description	Variables	Variable Description
$\alpha_{sa} \in \bar{A}$	source s to area a	fl_{at}	flow on arc λ_{at}
$\beta^{ki}_{ab} \in \bar{B}$	area a to pick-up b of trip k for resource i	fl^{ki}_{ab}	flow on arc β^{ki}_{ab}
$\gamma^{ki}_{bc} \in \bar{\Gamma}$	pick-up b to drop-off c of trip k for resource i	fl^{ki}_{bc}	flow on arc γ^{ki}_{bc}
$\delta^{ki}_{cb} \in \bar{\Delta}$	drop-off c to pick-up b of trip k to trip $k+1$	fl^{ki}_{ct}	flow on arc ε^{ki}_{ct}
	for resource i, for $k = 1, ..., K-1$	w^{1i}_{hb}	{**1**: if route on ζ^{1i}_{hb} selected, **0**: otherwise}
$\varepsilon_{ct} \in \bar{E}$	drop-off c to sink node t	x^{ki}_{bc}	{**1**: if route on γ^{ki}_{bc} selected, **0**: otherwise}
$\zeta^{1i}_{hb} \in \bar{Z}$	initial resource location h to pick-up b	y^{ki}_{cb}	{**1**: if route on δ^{ki}_{cb} 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
		z_i	{**1**: if resource i selected, **0**: otherwise}
		n_a	number of non-evacuated people at area a

$$
\begin{aligned}
s_i = & \sum_{\zeta^{1i}_{hb} \in \bar{Z}} \left(t^i_{hb} w^{1i}_{hb}\right) + \sum_{\gamma^{ki}_{bc} \in \bar{\Gamma}} \left(t^i_{bc} x^{ki}_{bc}\right) + \sum_{\delta^{ki}_{cb} \in \bar{\Delta}} \left(t^i_{cb} y^{ki}_{cb}\right) + \\
& \sum_{\zeta^{1i}_{hb} \in \bar{Z}} \left(u_i w^{1i}_{hb}\right) + \sum_{\zeta^{1i}_{hb} \in \bar{Z}} \left(o_i w^{1i}_{hb}\right) + \\
& \sum_{\delta^{ki}_{cb} \in \bar{\Delta}} \left(o_i y^{ki}_{cb}\right) + \sum_{\gamma^{ki}_{bc} \in \bar{\Gamma}} \left(p_i x^{ki}_{bc}\right) && \forall i \in I
\end{aligned}
$$

$$r \leq T$$

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

$$fl^{ki}_{bc} \leq q_i(x^{ki}_{bc}) \qquad \forall \gamma^{ki}_{bc} \in \bar{\Gamma}$$

$$fl^{ki}_{bc} \leq q_i(z_i) \qquad \forall \gamma^{ki}_{bc} \in \bar{\Gamma}$$

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

$$\sum_{\beta^{ki}_{aj} \in \bar{B}: j=b} fl^{ki}_{ab} = \sum_{\gamma^{ki}_{jc} \in \bar{\Gamma}: j=b} fl^{ki}_{bc} \qquad \forall b \in B, \forall k \in K, \forall i \in I$$

$$\sum_{\gamma^{ki}_{bj} \in \bar{\Gamma}: j=c} fl^{ki}_{bc} = fl^{ki}_{ct} \qquad \forall c \in C, \forall k \in K, \forall i \in I$$

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

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

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

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

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

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

$$fl_{at} \geq 0 \qquad \forall \lambda_{at} \in \bar{A}$$

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

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

$$fl_{ct}^{ki} \geq 0 \qquad \forall \varepsilon_{ct}^{ki} \in \bar{E}$$

$$s_i \geq 0 \qquad \forall i \in I$$

$$r \geq 0$$

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

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

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

$$n_a \geq 0 \qquad \forall a \in A$$

3 META-HEURISTIC SOLUTION METHOD SELECTION

A meta-heuristic method selection is developed to simplify the classification of the second-stage of the S-ICEP. However, a similar structure to the second stage of S-ICEP has not been observed in past studies. The closest related problem is the Bus Evacuation Problem (BEP) (Bish 2011) and its military variations (Dikas and Minis 2016), which solves a similar problem for on-land transportation using a homogeneous fleet of buses using a constructive heuristic. The approach provides a less complex solution since routes are interchangeable between resources without impact. A similar idea for a constructive heuristic has been applied to solve the ICEP by Krutein and Goodchild (2022). However, the solution quality of this approach varies significantly depending on which input data set is provided. It thus does not produce solutions with a predictable quality. The BEP has further been approximated through considering more restrictive branching methods that establish bounds in a quicker way than a commercial solver and solve to these (Goerigk et al. 2013). However, these problems neither show a performance guarantee and their performance is also highly dependent on the data set. The BEP in a simplified robust form has been successfully solved using a linear search and a tabu search approach with restarts (Goerigk and Grün 2014). Tabu searches are good algorithms to avoid getting trapped at local minima (Glover 1986). However, the presented linear and tabu search approaches, like the constructive heuristic presented by Bish (2011) take advantage of the strong symmetry of the problem in between the different resources by adding or deleting resources and/or trips from the route plans and/or re-allocating them to other resources and are based on the interchangeability of route components in between resources. Using a similar logic with a heterogeneous resource set leads to either many infeasible solutions due to compatibility issues or not exploring the entire feasible region since some solutions will never be generated through this approach and generating space efficient tabu lists that avoid revisiting recently visited solutions is difficult. Experiments with an example case of the S-ICEP have confirmed this.

The search for approximate solutions is therefore generalized to routing and combinatorial problems. Specifically for stochastic combinatorial problems, which the S-ICEP belongs to. Bianchi et al. (2009) have identified meta-heuristics that have shown to be successful in addition to tabu searches. These were ant colony optimization (Dorigo et al. 1999), simulated annealing (Kirkpatrick et al. 1983), and evolutionary/genetic algorithms (Goldberg 1989). While there have been some successful applications for these algorithms to stochastic combinatorial problems, especially when paired with warm start procedures and local searches, the size of the feasible space in combinatorial problems often requires a simplification of the solution representation to fit into memory for larger problem sizes and the performance of these meta-heuristics cannot be reliably predicted for arbitrary problems and requires analysis specific to the problem at hand (Bianchi et al. 2009).

For that reason, a specific genetic algorithm type is selected as a way to escape the dimensionality problem of solution representation through a soft representation, the Random-Key Genetic Algorithm (RKGA) (Bean 1994). Traditional genetic algorithms, which use a variable representation in the chromosome however, are not per default suitable to solve combinatorial problems, since the use of binary and integer variables gives little flexibility to alternate solutions in a local search without making them infeasible. RKGAs on the opposite represent the solution in a soft way, where, instead of every chromosome element representing a variable, the elements, which contain random keys between 0 and 1 represent parts of the physical problem behind that can be translated into a feasible solution to the problem on every iteration and stored in a simple list of continuous values (Bean 1994). For example, for a Traveling Salesman Problem (TSP), the nodes can be represented in a list, and the list indexes can represent the customers in the network. Using the random key values generated by the algorithm for every customer, the list of customers can be sorted in ascending order, which represents a feasible tour in the TSP if visited in that order (Bean 1994). Like in a regular genetic algorithm, during every evolution chromosomes from the population are combined and crossed with each other to generate off-spring. After ranking the fitness of the available solutions in the population, the best solutions (e.g. top 10%) are determined the elite set. To enhance diversity and improve the solution over time, the worst solutions (e.g. worst 10%) are dropped from the population and replaced by random new chromosomes in every iteration. The RKGA will eventually find the optimal solution in expectation based on survival-of-the-fittest. Extensions of this methodology that improve the performance further for many problem types have been to bias towards the set of elite solutions in every evolution step (every off spring is generated from at least one elite-set solution) through Biased Random-Key Genetic Algorithms (BRKGA) (Gonçalves and Resende 2011). The BRKGA has been further improved through using multiple parents to generate offspring instead of just two (MP-BRKGA) (Andrade et al. 2021), which has proven to improve the performance of the regular BRKGA. This algorithm was chosen as its soft representation structure appears suitable to store the complexity of the S-ICEP problem, and the introduced randomness allows for exploring multiple areas across the feasible region and thus, escape local optima. An additional benefit of RKGA type algorithms that informed its selection as the method of choice is its suitability for parallel processing. Since in every evolution, all chromosomes have to be evaluated independently of each other, parallel processing can be a powerful tool that supports accelerating the algorithm. Since the only problem dependent part of the MP-BRKGA is the decoder function that translates the chromosome into a feasible solution to the S-ICEP, the main objective of this paper is therefore the development of a decoder logic that suits the S-ICEP and its validation in computational experiments using the MP-BRKGA framework provided by Andrade et al. (2021).

4 DECODER DESIGN

To design a decoder for the S-ICEP to be used in the MP-BRKGA, it is necessary to represent a solution through a single list of values between 0 and 1. The structure presented in Algorithm 1 was most successful in generating good route plans for the S-ICEP. First, generate a list of all pick-up and drop-off points, add an empty row (representing no node visit) and map them to a sequence of equally spaced threshold points between 0 and 1 (e.g. if there are 3 points in the list plus the empty one, the mapping would be $[0, 0.25, 0.5, 0.75, 1]$. For Ξ scenarios, I resources, and K maximum trips per resource, the chromosome will have the length: $\Xi * I * K$. Every iteration of the MP-BRKGA then generates a chromosome of values between 0 and 1. The procedure presented in Algorithm 1 then translates this chromosome into a feasible solution to the S-ICEP.

Algorithm 1: MP-BRKGA Decoder for S-ICEP.

Result: A feasible, near-optimal evacuation route plan

Initialize all resources $i \in I$, evacuees $e \in E$ as the evacuees, pick-up nodes $b \in B$, drop-off nodes $c \in C$, scenarios $\xi \in \Xi$;

Split the chromosome into Ξ parts of equal length $I * K$ (each representing the solution for one scenario ξ);

for *every chromosome segment* $\xi \in \Xi$ **do**

    Split each sub-chromosome into I parts of length K (each representing the route of a specific resource i in scenario ξ). **for** *every sub-chromosome segment* $i(\xi) \in \xi$ **do**

        Map the values between

$$0, 1$$

        at each index to the appropriate pick-up or drop-off point, based on threshold-based mapping;

    end

    Aggregate all route segments of all resources $i(\xi) \in \xi$ into a sortable list;

    Order the list of route segments by their respective arrival time;

    Allocate the evacuees at each pick-up point b to the resources i based on the order of resource arrivals, considering the respective capacity of every resource;

    Remove all route segments at each pick-up point after the last route segment that had passengers allocated. These trips do not need to be executed, as they do not carry passengers;

    Calculate route length s_i for each resource i, and denote the longest one as r;

    Calculate the operational cost for each resource i, and denote the longest one as r;

    Calculate the number of evacuees left behind n_a at each location a, if any remain;

end

For every scenario ξ, aggregate the values for $r(\xi)$, $\sum_{i \in I} cv_i * s_i(\xi)$, and the persons left behind $n_a(\xi)$;

Calculate the fitness of the solution through the S-ICEP objective, considering the relative probabilities for each scenario.

Figure 1 illustrates this decoder mechanism for an example. The design of this decoder allows that the continuous values generated by the MP-BRKGA can be decoded and used to generate feasible solutions to the S-ICEP in an efficient manner. This problem structure ensures that the entire feasible region is explored. Since the goal of this paper is to identify and develop a method that can beat the time performance of a commercial solver while providing more reliable solution performance than the constructive heuristic presented by Krutein and Goodchild (2022), comparative experiments are presented in the following section.

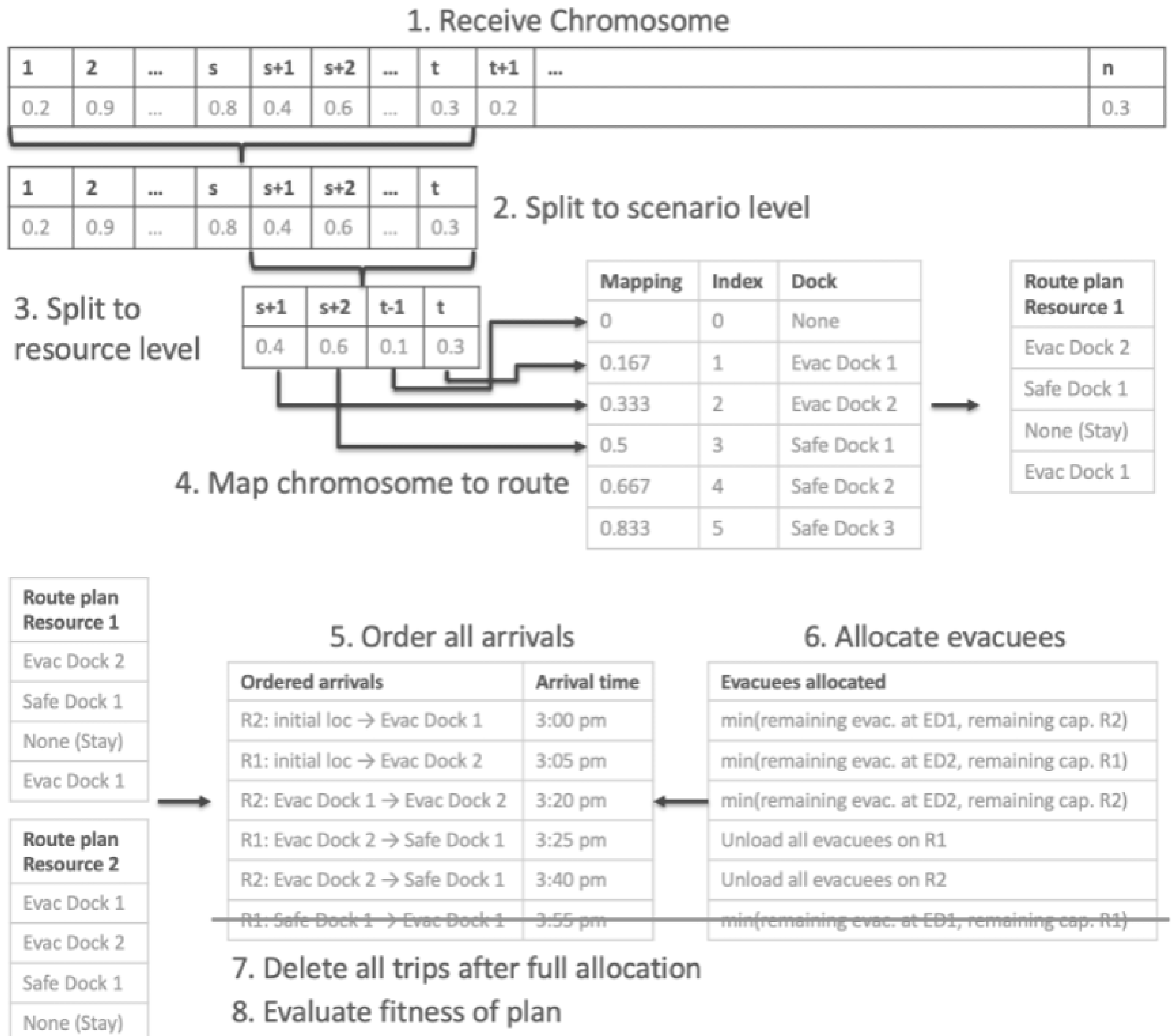

Figure 1: Example visualization of decoder translation from chromosome to feasible solution of S-ICEP.

5 EXPERIMENT RESULTS

Experiments were conducted to investigate whether the MP-BRGKA with the decoder developed above can beat the performance of an exact commercial solver. As a benchmark, the S-ICEP was implemented in a formal mathematical structure using the Pyomo modeling environment in Python 3.9, using the commercial Gurobi 9.0 solver. The BRKGA was implemented using the Python MP-BRKGA package designed by Andrade et al. (2021). Consequently, the decoder was also implemented using Python, in an object-oriented structure. The experiments on the BRKGA were conducted in a concurrent and parallelized version. In addition, a random data generator was used for the S-ICEP with inputs on the number of scenarios, number of candidate resources, and the number of nodes. Using these parameters, the problem is then randomly generated using a set of uniform and gamma distributions, depending on the parameter at hand to keep the problems realistic (e.g. avoiding unrealistic examples where a resource has a capacity of 100,000 passengers). The MP-BRKGA maintains a solution population size proportional to the problem size itself and is determined through the following formula to keep enough chromosomes in the population to maintain a large enough level of variety: Population size $= \Xi * I * K$. Key metrics on the test data sets are presented in Table 2. For the study presented in this paper, a total of 5 randomly generated experiments have been conducted. The experiments were conducted on a cloud computing instance with 36 CPUs and 72 GB of RAM, and were aborted after a maximum run time of 1,800s, and a maximum evolution of 1,000 generations in the MP-BRKGA. The results are presented in Table 3. For Gurobi 9.0, only solutions from the concurrent run mode are displayed, as the parallelized run mode performed significantly worse due to increased computational overhead.

Table 2: S-ICEP Test Data Sets.

	Test 1	Test 2	Test 3	Test 4	Test 5
Sets			*Set size*		
Scenarios	2	2	2	3	4
Potential Resources	6	4	2	5	20
Docks	7	5	5	8	6
Round trips	2	2	2	3	4
Parameters			*Setting*		
Penalty	5,000	5,000	5,000	5,000	5,000
Variable Type			*Quantity*		
Continuous Variables	416	550	650	8308	17,290
Binary Variables	413	595	758	12,875	24,820

Table 3: Experiment Results Gurobi 9.0 vs. BRKGA Decoder.

Data	Gurobi 9.0 (Concurrent)		BRKGA (Concurrent)		BRKGA (Parallelized)	
	Solution time	Objective	Solution time*	Objective	Solution time*	Objective
Test 1	5.51s	101.03	109.77s	172.00	142.41s	124.00
Test 2	2.36s	56.67	188.13s	56.67	17.65s	56.67
Test 3	116.15s	229.00	375.28s	324.00	928.2s	232.64
Test 4	1,800.00s	**313.04	805.57s	291.39	671.39s	259.73
Test 5	1,800.00s	**178.04	1,217.39s	218.25	908.63s	108.03

*Last improvement

**Runs aborted after 1,800s; best available solution displayed.

Reviewing the results presented in Table 3, MP-BRKGA was not able to beat Gurobi 9.0 in solution performance in all test instances. Only in Test 2 it was able to reach the objective value Gurobi was able to find, but in a much longer time frame than Gurobi, both for the concurrent and parallelized version of the BRKGA. However, for the larger instances, the performance of the MP-BRKGA in comparison to Gurobi

9.0 is better, particularly for the parallelized version. While the methods that Gurobi is using guarantee convergence to a global optimum, the solution time becomes so long that the optimal solution cannot be reached within a reasonable time frame for emergency management. For these cases, the parallelized MP-BRKGA can be useful to obtain solutions in a more reasonable time frame. The underlying reasons and implications of these experiment results are discussed in the following section.

6 DISCUSSION OF RESULTS

The results from the experiments have shown that the implemented version of the MP-BRKGA has outperformed the Gurobi implementation for larger problem instances. This makes the method useful for emergency use, since obtaining solutions quickly is instrumental during situations where human lives are in danger. While the solutions generated by the parallelized decoder are fine solutions to the problem at hand, the algorithm speed could still be improved for smaller instances. Investigating reasons for this lack of performance, the first disadvantage that can be noted is that, the MP-BRKGA only approximates optimal solutions due to its iterative and randomized nature. The main problem in the decoder structure is that the feasible region of the S-ICEP is so large that the solution space that is searched by the decoder causes a lot of sub-optimal solutions to be generated. Even though this ensures a diversified set of chromosomes, it does not support the progression to a better solution. While it is difficult to develop meta-heuristic methods with performance guarantees, further efforts could be placed on letting the decoder bias towards solutions that have a high likelihood to improve the set of solution candidates, while reducing the risk that the global optimum is accidentally excluded.

Furthermore, the algorithm needs to evolve through many generations to receive solutions with good fitness values. This means that the speed of the decoder function is crucial in determining a short solution time. Unfortunately, the use of an object-oriented representation of the problem requires the use of for loops in Python. Since Python is a scripting language and therefore slow processing of for loops (lack of memory allocation, no full multi-threading), it is difficult to beat commercial solvers that run on faster programming languages.

7 CONCLUSIONS AND NEXT STEPS

In this paper, a new decoder logic was presented to solve the S-ICEP problem through a MP-BRKGA. The decoder function is well suited to generate high quality solutions to the S-ICEP problem in a reasonable time frame and thus provides an important contribution in identifying a faster way to solve the S-ICEP. This allows emergency personnel, planners and coordinators to make routing decisions that resolve a dangerous situation quickly and therefore contributes to national security. This is particularly the case for larger problem instances with a large number of evacuees, a large number of pick-up and drop-off points and high heterogeneity in the data set. One example are evacuations of large isolated areas affected by natural or human-made hazards such as fires or nuclear reactor leaks. Another example are evacuations triggered by military conflicts, such as the evacuation of encircled/isolated troops or civilians that have to be evacuated by air or by sea from multiple locations through a coordinated effort. Furthermore, islands that require evacuation by the coast guard due to tsunami or wildfire hazards.

The speed of the current implementation of the algorithm allows for improvements. One part of these improvements could be dedicated towards giving the decoder more bias to high quality solutions and thus decrease the number of generations that need to be evolved to get close to the global optimum. It could be considered to reduce the feasible region of the decoder to only generate solutions from regions that are promising to contain good route plans. For example, if a certain number of resources are being used, it is likely that a good solution will try to keep all available resources busy at all times to shorten the evacuation time, thus solutions that do not keep all resources busy could be discarded. Another direction with similar effect could be to make the decoder two-staged and implement a high level quick approximation of the objective value in the first step and only evaluate the objective function exactly if a certain threshold is reached to reduce the computational effort. This way, not much time is wasted on finding the actual objective value of solutions that are not promising anyways. A combination of both approaches can also be considered. Furthermore, future work can further compare the computational performance of the MP-BRKGA approach from this paper to the previously proposed structural heuristic (Krutein and Goodchild 2022), in addition to Gurobi.

Another part of these considerations is implementation based and can be achieved through choosing a different way to implement this algorithm. In that sense, the next steps become more of an implementation challenge than a modeling or design challenge. A vectorized problem representation in the decoder has the potential to significantly accelerate the algorithm and should be considered to improve the solution quality.

Lastly, another approach would be to choose an entirely different solution approach to the S-ICEP. Promising research directions are the decomposition of the problem using column-generation methods that separate the problem into a route generation problem and an evacuee allocation problem (Lübbecke and Desrosiers 2005). Another promising approach is the use of deep learning, which was recently explored for dynamic programs in inventory replenishment (Qi et al. 2021). The main idea is to predict the model outputs directly from input data after training a deep learning model from a training data set of problem instances solved to the optimum using a commercial solver.

ACKNOWLEDGEMENTS

The work presented in this paper has been performed in the context of the project ‘Shipping Resilience: Strategic Planning for Coastal Community Resilience to Marine Transportation Risk (SIREN)’. This project is financially supported by the Marine Observation, Prediction and Response (MEOPAR) Network of Centres of Excellence (NCE) under the Award Number 2-02-03-041, and by the Province of British Columbia. This funding source did not provide any support in study design; in the collection, analysis and interpretation of data; in the writing of the report; and in the decision to submit the article for publication. This financial support is gratefully acknowledged.

REFERENCES

ABC News 2020. “Mallacoota Evacuations Begin as Thousands Trapped by Bushfires Are Transported to Navy Ships”. *Australian Broadcasting Corporation News*. https://www.abc.net.au/news/2020-01-03/navy-starts-bushfires-evacuations--in-mallacoota/11838424, accessed 10th February 2022.

Andrade, C. E., R. F. Toso, J. F. Gonçalves, and M. G. Resende. 2021. “The Multi-Parent Biased Random-Key Genetic Algorithms With Implicit Path-Relinking and its Real-World Applications”. *European Journal of Operational Research* 289(1):17–30.

Bean, J. C. 1994. “Genetic Algorithms and Random Keys for Sequencing and Optimization”. *ORSA Journal on Computing* 6(2):154–160.

Bianchi, L., M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. 2009. “A Survey on Metaheuristics for Stochastic Combinatorial Optimization”. *Natural Computing* 8(2):239–287.

Bish, D. R. 2011. “Planning for a Bus-Based Evacuation”. *OR Spectrum* 33(3):629–654.

Britten, L. 2019, Mar. “How Do You Evacuate All of Bowen Island? Municipality Seeks Plan to Find Out”. *CBC News*. https://www.cbc.ca/news/canada/british-columbia/bowen-island-plan-1.5059389, accessed 2nd August 2021.

Coffey, H. 2019, Aug. “Greece Wildfires: Hundreds of Tourists Evacuated from Hotels and Beaches”. *The Independent*. https://www.independent.co.uk/travel/news-and-advice/greece-wildfires-tourists-evacuated-beach-hotel-samos-island-fire-a9080281.html, accessed 2nd August 2021.

Dikas, G., and I. Minis. 2016. “Solving the Bus Evacuation Problem and its Variants”. *Computers and Operations Research* 70:75–86.

Dorigo, M., G. Di Caro, and L. M. Gambardella. 1999. “Ant Algorithms for Discrete Optimization”. *Artificial Life* 5(2):137–172.

Glover, F. 1986. “Future Paths for Integer Programming and Links to Artificial Intelligence”. *Computers and Operations Research* 13(5):533–549.

Goerigk, M., and B. Grün. 2014. “A Robust Bus Evacuation Model with Delayed Scenario Information”. *OR Spectrum* 36(4):923–948.

Goerigk, M., B. Grün, and P. Heßler. 2013. “Branch and Bound Algorithms for the Bus Evacuation Problem”. *Computers and Operations Research* 40(12):3010–3020.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Gonçalves, J. F., and M. G. Resende. 2011. "Biased Random-Key Genetic Algorithms for Combinatorial Optimization". *Journal of Heuristics* 17(5):487–525.

Kirkpatrick, A. S., C. D. Gelatt, M. P. Vecchi, S. Science, N. Series, and N. May. 1983. "Optimization by Simulated Annealing". *Science* 220(4598):671–680.

Krutein, K. F., and A. Goodchild. 2022. "The Isolated Community Evacuation Problem with Mixed Integer Programming". *Transportation Research Part E: Logistics & Transportation Review* 161(102710).

Krutein, K. F., J. McGowan, and A. Goodchild. 2022. "Evacuating Isolated Islands with Marine Resources: A Bowen Island Case Study". *International Journal of Disaster Risk Reduction* 72(102865).

Lübbecke, M. E., and J. Desrosiers. 2005. "Selected Topics in Column Generation". *Operations Research* 53(6):1007–1023.

Qi, M., Y. Shi, Y. Qi, C. Ma, R. Yuan, D. Wu, and Z. M. Shen. 2021. "A Practical End-to-End Inventory Management Model with Deep Learning". *SSRN*.

AUTHOR BIOGRAPHIES

KLAAS FIETE KRUTEIN is an Operations Research Scientist. He holds a Ph.D. and a M.S. awarded by the Department of Industrial & Systems Engineering at the University of Washington. His research interests include optimization under uncertainty, global optimization, combinatorial optimization, vehicle routing, and applications of data-driven optimization and simulation techniques in logistic and supply chain fields. He is a member of INFORMS. His e-mail address is kfkru@uw.edu. His website is https://singfie.github.io.

ANNE GOODCHILD is a Professor in the Department of Civil & Environmental Engineering at the University of Washington and is the Founding Director of the Supply Chain Transportation & Logistics Center. She holds a Ph.D. degree in Civil and Environmental Engineering from the University of California, Berkeley. Her email address is annegood@uw.edu. Her website is https://depts.washington.edu/sctlctr/about-us/faculty-professional-staff/dr-anne-goodchild.

LINDA NG BOYLE is a Professor in the Department of Industrial & Systems Engineering at the University of Washington. She has a joint appointment in Civil & Environmental Engineering linda@uw.edu. She has a Ph.D. and M.S. from the University of Washington and her research focus is on human factors and modeling operator behavior. Her email address is linda@uw.edu. Her website is https://depts.washington.edu/hfsm/.