Bus Routing for Emergency Evacuations: The Case of the Great Fire of Valparaiso

Javiera Loyola Vitali*, María-Cristina Riff† and Elizabeth Montero‡
Department of Computer Science
Universidad Técnica Federico Santa María
Valparaíso, Chile
Email: *Javiera.Loyola@inf.utfsm.cl, †Maria-Cristina.Riff@inf.utfsm.cl,
‡Elizabeth.Montero@inf.utfsm.cl

***Abstract*—The Bus Evacuation Problem is a route planning problem, in the context of an evacuation in an emergency situation. Considering that public transport is available to support the evacuation, the objective of the problem is to determine the best route for each vehicle, to move all the people from a risk zone to open shelters located in safe zones, such that the evacuation time is minimized. In this work we present a method based on the Greedy Randomized Adaptive Search Procedure metaheuristic to solve the problem, in order to apply the solution to a real-world scenario based on a recent wildfire on Valparaíso, Chile. In computational experiments we demonstrate that our approach is effective to solve real-world size problems, and able to outperform a commercial MIP solver.**

I. Introduction

The human kind has always faced natural threats, such as earthquakes, hurricanes or tsunamis, and human caused catastrophes, like wildfires or war events. These kind of situations require a mass evacuation, which needs a great amount of complex planification in order to minimize the number of affected victims. Most of the great scale evacuation research has been focused in car-based evacuation, but Hurricane Katrina, in 2005, highlighted the need to focus on transit-dependent residents, which were the most affected for not being able to leave the zone on their own means [5]. The Bus Evacuation Problem consists in the planification of a great scale evacuation on an endangered zone, previous to a catastrophic event. The goal is to transport the residents that inhabit the risk zone, to different shelters enabled for the emergency, in the most efficient manner.

The problem was first formulated by Bish et al [2]. Its purpose is the planification of the schedule of a series of buses that will perform the evacuation. The scenario considers that people inside a risky zone gather on different points distributed among the area. Each bus schedule is constructed by selecting the routes the buses have to travel, in order to iteratively pick people from the collection points and take them to one of the open shelters, considering that buses and shelters have capacity constraints. The objective is to find the planification that minimizes the evacuation time.

This problem can be considered as a variant of the Vehicle Routing Problem, including new elements that add more complexity. This, the bus evacuation problem is NP-complete, as shown in [12].

In this work, we propose a method based on the Greedy Randomized Adaptive Search Procedure metaheuristic to solve the problem. This work is organized as follows: In Section II we describe the problem in detail and present its mathematical formulation. In Section III, some related work on the bus evacuation problem, along with some variants, are presented. Section IV presents our proposal to solve the problem, and Section V describes the experiments we performed to test the proposed approach and the obtained results. Finally, on Section VI some conclusions and future work are presented.

II. Problem Description

The Bus Evacuation Problem (BEP) [2] consists in the planification of a large scale evacuation. In an emergency situation, such as flooding, tsunamis or wildfires, it's necessary to evacuate a large zone with the help of public transportation. This kind of evacuation is mainly intended for transit dependent people, which cannot leave the zone by themselves due to different mobility problems.

An evacuation scenario is composed by the following elements:

- **Evacuees:** group of people that require mobilization to leave a hazardous zone. The total number of evacuees is assumed known.
- **Buses:** the public transport vehicles that will move the evacuees. Each bus has a maximum capacity, considered equal for all vehicles.
- **Collection Points:** group of points distributed throughout the endangered zone, where the buses arrive to pick people up.
- **Shelters:** group of locations outside the endangered zone, where the evacuees will be delivered by the buses.
- **Yards:** points where the buses are located at the beginning of the evacuation process.

Collection points, shelters and yards, as well as the distances between them, can be represented in the form of a graph, showing which nodes are connected. Figure 1 shows a BEP graph example, with one yard, three collection points, and two shelters.

† Supported by FONDECYT Project no. 1151456. Partially supported by the Centro Científico Tecnológico de Valparaíso (CCTVal) no. FB0821
‡ Supported by FONDECYT Initiation into Research Project no. 11150787

978-1-5090-4601-0/17/$31.00

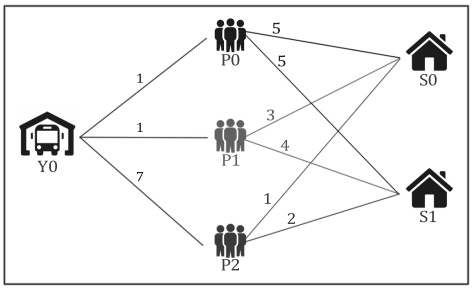

Fig. 1. Example of a BEP network graph.

A. Mathematical Model

In this work, we consider a simplified version of the model presented in [2]. The mathematical formulation is as follows.

1) Notation: We consider a network (N, A), where N and A represent sets of nodes and arcs, respectively. The set N includes three types of nodes: Y, P and S. Y is the set of yard nodes, where the buses are located at the beginning of the evacuation. These are the starting points of the routes. P is the set of collection points, where the evacuees will gather to take the bus. S corresponds to the set of shelters, where the evacuees will be dispatched from the buses. V is the set of available buses, each one with a capacity of Q. The buses are divided in subsets V_y, $y \in Y$, where each bus $v \in V_y$ is initially located in yard y. Each node $p \in P$ has a demand of D_p people waiting for a bus, and each shelter $s \in S$ has a capacity of C_s. Both values are known in advance. Each arc $(i, j) \in A$ has a travel cost $\tau_{ij} \geq 0$.

2) Decision variables:

- $x_{ij}^{vt} = \begin{cases} 1 & \text{If the bus } v \text{ uses the arc } (i,j) \text{ on trip } t \\ 0 & \text{otherwise} \end{cases}$
- $b_j^{vt} \geq 0$: Total number of evacuees on node j that get on the bus v after trip t, $\forall j \in N$, $v \in V, t = 1, ..., T$ (if j is a shelter, the number of evacuees that will leave the bus).
- T_{evac} : Total duration of the evacuation.

3) Objective: Minimize the duration of the evacuation. This is, minimize the time of the bus that takes the longest to complete its evacuation schedule (min-max function).

$$Minimize\, T_{evac} \tag{1}$$

4) Constraints: The minimization is subject to the following:

The total evacuation time must be greater or equal than the evacuation time of any bus.

$$T_{evac} \geq \sum_{(i,j)\in A} \sum_{t=1}^{T} \tau_{ij} x_{ij}^{vt}, \forall\, v \in V \tag{2}$$

Flow-balance constraint for collection points: A bus that arrives to node j on trip t, leaves the same node at trip $t+1$.

$$\sum_{i:(i,p)\in A} x_{ip}^{vt} = \sum_{k:(p,k)\in A} x_{pk}^{v(t+1)}, \quad \forall p \in P,\, v \in V,\, t = 1, ..., T-1 \tag{3}$$

Flow-balance constraint for shelter nodes: A bus can stay on the shelter s in its last trip.

$$\sum_{s:(s,j)\in A} x_{sj}^{vt} \geq \sum_{k:(s,k)\in A} x_{sk}^{v(t+1)}, \quad \forall s \in S,\, v \in V,\, t = 1, ..., T-1 \tag{4}$$

A bus can only make one trip at a time.

$$\sum_{(i,j)\in A} x_{ij}^{vt} \leq 1, \forall v \in V, t = 1, ..., T \tag{5}$$

Each bus must leave the yard in its first trip.

$$x_{yj}^{v1} = 1, \forall y \in Y,\, j : (y, j) \in A, v \in V \tag{6}$$

If a bus doesn't leave the yard at the beginning of the evacuation, it cannot leave later (it does not participate in the evacuation process).

$$x_{yj}^{vt} = 0, \forall y \in Y, j : (y, j) \in A,\, v \in V, t = 2, ..., T. \tag{7}$$

The last trip of a bus cannot finish in a collection point.

$$x_{ip}^{vT} = 0, \forall p \in P, i : (i, p) \in A, v \in V \tag{8}$$

A bus can only pick up evacuees from node j if it has traveled to that node.

$$b_j^{vt} \leq \sum Q x_{ij}^{vt}, \forall j \in N,\, v \in V, t = 1, ..., T \tag{9}$$

Bus capacity must not be exceeded.

$$0 \leq \sum_{j\in P} \sum_{l=1}^{t} b_j^{vl} - \sum_{k\in S} \sum_{l=1}^{t} b_k^{vl} \leq Q,\, \forall v \in V, t = 1, ..., T \tag{10}$$

Shelter capacity must not be exceeded.

$$\sum_{m\in V} \sum_{t=1}^{T} b_j^{mt} \leq C_j, \forall j \in S \tag{11}$$

All evacuees must be picked up from the collection points.

$$\sum_{m\in V} \sum_{t=1}^{T} b_j^{mt} = D_j, \forall j \in P \tag{12}$$

All evacuees must be moved to the shelters.

$$\sum_{p\in P} \sum_{t=1}^{T} b_p^{vt} = \sum_{s\in S} \sum_{t=1}^{T} b_s^{vt}, \forall v \in V \tag{13}$$

5) Additional considerations:

- The duration of one trip is considered proportional to the cost of that trip, as well as the distance. Therefore, the terms "time", "cost" and "distance" are used indistinctly.
- All the costs in the network graph are symmetric, i.e. $\tau_{ij} = \tau_{ji}$
- The goal is to schedule the evacuation route of each bus, minimizing the duration of the evacuation, while covering all the demand, without breaking the capacity constraints of shelters and buses.

III. Related Work

The Bus Evacuation Problem was first proposed and formalized by [2], which presented two mixed-integer programming formulations for a bus-based evacuation planning. In its work, the author establishes a comparison between the BEP and the Split Delivery Multi-Depot Vehicle Routing Problem (SDMD-VRPI), a variant of the Vehicle Routing Problem that combines the Multi-Depot Vehicle Routing Problem with Inter-depot routes (MDVRPI) [3], and the Split Delivery VRP (SDVRP). Although the BEP and the SDMDVRPI are essentially very similar, some of the main differences between them are the objective and the network structure. Two objective functions are analyzed: the min-max function, which minimizes the evacuation duration (minimizes the duration of the bus with the longest duration); and the cost function, that minimizes the total routing cost of all buses. Results using MIP solver CPLEX demonstrated that the min-max function is harder to solve, but with the use of heuristics it performs better for bigger instances.

In [13], a simplified version of the scenario in [2] is considered. Various approaches to find lower and upper bounds are proposed, which are used to solve the problem with a branch and bound framework, that includes different ramification and pruning rules. For the experiments, randomly generated instances are solved using the proposed framework, along with CPLEX. The results show that all of the proposed techniques are faster than the MIP solver, and for the lower bounds, the better approach is to minimize the sum of the travel times.

In [12], the Robust BEP is presented. The exact number of evacuees on each collection point is unknown, but some possible scenarios are given. The problem is to decide for each bus if it's better to evacuate immediately, considering this uncertainty (here-and-now buses), or wait until all evacuees arrive to the collection point and the exact number is known (wait-and-see buses). A MIP formulation based on [2] is presented, and solved using CPLEX, linear search with CPLEX, and a tabu search based framework, along with some strategies to calculate lower bounds. Randomly generated instances and a real world based scenario in Kaiserslautern, Germany, are used for experiments. Results show that for the bigger instances, tabu search is the only one producing solutions, although linear search also works for the real world scenario. Data uncertainty is also considered in [15], which presents a model for evacuation, and propose algorithms based on hybrid genetic algorithms, artificial neural network and hill climbing to solve it. In [19], the BEP and the RBEP models are applied to a real world scenario in Nepal, and solved using the branch and bound approach presented in [13], and the tabu search heuristic from [12], respectively. Results show the relevance of perfect information on the evacuation performance, having that the RBEP produces solutions around 30 minutes more than the BEP. Also, they conclude that the most relevant element on the evacuation performance is the bus capacity, having that less big buses would be better than more small buses for a faster evacuation.

Another variant is the Integrated BEP (IBEP) [14], which includes the allocation of collection points and shelters. To solve this, a Branch-Cut-and-Price approach is proposed. The results on randomly generated instances show that the proposed algorithm is able to find the optimal value in more instances than CPLEX, and finds smaller gaps ($(UB - LB)/LB$) in general. Finally, the real world scenario from [12] is solved with the BCP approach, being able to find solutions for a real size problem instance.

In [11], the Comprehensive Evacuation Problem (CEP) considers the route planning, as well as the shelter allocation, taking into account the interaction between public and private transit, and the risks of the evacuation. A NSGA-II algorithm is proposed for this multiobjective model, minimizing the evacuation time, the risk of the evacuation, and the number of used shelters. The resulting solutions are later improved using local search. Data aggregation techniques are also discussed, in order to generate solutions for real world scenarios starting from a reduced solution. For the experiments, two real-world based instances are used, concluding that the genetic algorithm is effective to solve the multi-criteria variant. The aggregation technique allows finding solutions for big instances, by reducing them and, therefore, taking less time to solve them. Another multiobjective evacuation model is considered in [1], modeling private transit with an optimal spatiotemporal evacuation (OSTE) formulation, and mass public transportation with a multiple-depot, time-constrained, pickup and delivery vehicle routing problem (MDTCPD-VRP).

A survey on evacuation problems modeled as network-flow graphs is presented in [4]. Small and large scale evacuations are considered. Due to the specific features of each model, it is very hard to unify the proposals, or identify the better strategy. In the context of large scale evacuation, it is mentioned that most of the proposed models fail to incorporate psychological aspects of evacuees, so they cannot be yet directly applied to real situations. Other reviews covering high-way based evacuations and humanitarian logistics can be found in [17] and [18], respectively.

IV. Algorithm Proposal

In this work we propose a Greedy Randomized Adaptive Search Procedure (GRASP) [6], [7], [8], [9], [10] based algorithm. GRASP is an iterative metaheuristic, consisting of three main phases: pre-processing, construction and local search. At each iteration, a feasible solution is built using a non-deterministic greedy, and improved in the local search phase. This process is carried out until some termination criterion is reached. An additional phase for initialization can be performed to make some initial processing, before the iterations start.

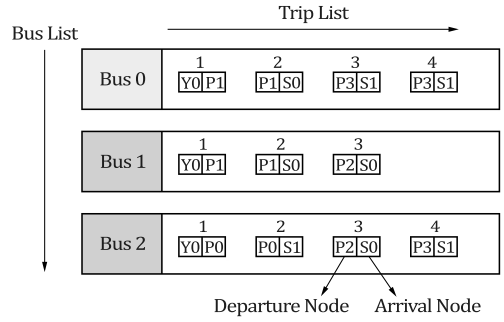

Fig. 2. Solution representation.

A. Representation

The solution is represented using an array, where each index corresponds to one of the buses. Each element points to a list of trips, which corresponds to the complete route schedule for the respective bus. Each trip is represented as a pair (i, j), where i corresponds to the departure node, and j is the arrival node. To facilitate the manipulation of the solution, the representation is composed of two types of trips: an initial trip, which goes from a yard to a collection point and does not move passengers; and one or more evacuation trips, traveling from a collection point to a shelter. A move from a shelter to a collection point is not considered as a trip, but inferred from two connected trips instead.

Figure 2 shows a solution representation example, for an evacuation with three buses, one yard, four collection points and two shelters. For example, for bus V_1 we have the route $\{(P_1, S_0), (P_2, S_0)\}$. This means that the bus went from collection point P_1 to shelter S_0, then went to P_2 and returned to S_0. The move from S_0 to P_2 is called *return trip*, and its unique purpose is to consider the cost of that trip. This representation allows us to know and evaluate quickly the route assigned to a bus, as well as an easy manipulation of the trips, in order to perform changes to a candidate solution. Additionally, this approach considers only feasible solutions, incorporating many of the problem constraints into the design of the solution.

B. Initialization Phase

This phase is performed only once, at the beginning of the procedure. First, we identify all the possible trips to perform on the graph, i.e. the arcs, and a Trip object is saved for each connection. A difference is stablished between an *Initial Trip* objects, which represents the connection between a yard with a collection point, and a *Trip* object, which connects a collection point with a shelter. After that, a list of the feasible trips is assigned to each node, this is, the trips that a bus can perform from that node. The shelter nodes also get a list assigned, with the trips that arrive to each one.

C. Constructive Phase

In this phase, a non-deterministic greedy is used. As the objective function is to minimize the total distance traversed by the bus with the longest route, the construction rule consists in selecting the nearest node that can be visited from the current node, that is, choosing the shorter feasible trip. The main idea is to reduce the total length of each route, in order to obtain a better value for the objective function. The restricted candidate list (RCL) consists of the k shortest trips from each node that were generated on the Initialization phase. The size of the RCL is controlled by the parameter $\alpha \in [0, 1]$.

Algorithm 1 shows the pseudocode of the Constructive Phase.

Algorithm 1: GRASP Constructive phase.

Input: An instance I of the problem, RCL size $\alpha \in [0, 1]$

```
1  PerformInitialTrip();
2  while evacuation not completed do
3      foreach bus in V do
4          curr ← GetCurrentPosition(bus);
5          T ← GetFeasibleTripsFromNode(curr);
6          rclSize ← α∗Size(T);
7          trip ← SelectRandomTrip(T, 0, rclSize);
8          if curr ∈ P then
9              AddTrip(bus, trip);
10             curr ← ArrivalNode(trip);
11             UpdatePosition(bus, curr);
12             capacity ←DecreaseCapacity(curr);
13             if capacity == 0 then
14                 foreach p in P do
15                     RemoveTripByArrivalNode(p, curr);
16                 end
17             end
18         else                                   /∗curr ∈ S∗/
19             curr ← ArrivalNode(trip);
20             UpdatePosition(bus, curr);
21             demand ←DecreaseDemand(curr);
22             if demand == 0 then
23                 foreach s in S do
24                     RemoveTripByDepartureNode(s, curr);
25                 end
26             end
27         end
28     end
29 end
```

At the beginning of a GRASP iteration, all buses are in their respective yards. All evacuees are reunited at the collection points, and shelters are empty. The initial trips are selected randomly from the RCL of each bus's starting node. Each selected trip is then added to the schedule of each bus, and the bus position updated to the arrival node. Once the bus *arrives* to the collection point, the node's demand decreases by an amount equal to the bus capacity (people get on the bus). This, under the model assumption that the demand on each collection point is a multiple of the capacity of the buses. Once the bus has picked up the passengers, a new trip is selected from the node's RCL, added to the bus schedule, the bus location updated, and the arrival shelter's capacity decreased. Now that the bus is located at a shelter, a new trip is chosen from the RCL to return to a collection point. The bus location is updated, the arrival collection point's demand updated, but the trip is not added to the bus schedule.

Every time a bus arrives to a collection node/shelter, the demand/capacity are immediately updated, so the other buses' RCL will not consider unfeasible trips. The procedure is performed until there is no people left on the collection points.

D. Local Search Phase

To improve the solution resulting from the Constructive Phase, a first improvement Hill Climbing has been implemented. The representation of the solution is the one described on Section IV-A. On each iteration of HC, we select the first neighbor solution that improves the evacuation time.

To generate the neighborhood, a shift move is proposed. Algorithm 2 shows a pseudocode of the procedure. The first step is to identify the bus with the longest schedule (in terms of distance traversed). After that, we remove each trip with the form (P_i, S_j) from the selected bus, and try to insert it into every position of all the other buses' trip list. This is only performed until we find the first shift that produces a solution with a better objective function. The proposed move creates feasible solutions. If an assigned trip (P_i, S_j) is deleted from the schedule of the bus performing it, the evacuees transported in that trip will simply take another bus (the recipient bus of the shift) in another time of the evacuation, and the trip always will go from collection point P_i to shelter S_j. The HC algorithm is performed until a maximum number of iterations is reached, or until some local optimum is reached.

Algorithm 2: Shift move for HC

```
Input: A solution C of the problem
Output: Bool of success
Best ← GetOFValue(C);
bMax ← GetLargestRouteBus(C);
foreach t in GetTripList(bMax) do
    RemoveTripFromBus(t, bMax);
    foreach b in V; b ≠ bMax; V ∈ C do
        foreach pos in GetTripList(b) do
            InsertTripInPosition(t, pos);
            if GetOFValue(C) < Best then
                return true
            else
                UndoTripInsertion(t, pos);
            end
        end
    end
    UndoTripRemoval(t, bMax);
end
return false
```

V. Experimental Results

In this section, we test the capability of the proposed approach to solve a real world based scenario, as well as its performance and quality solving instances of different sizes. For this, we present two datasets, and describe the performed experiments.

A. Datasets

Two sets of instances have been generated for the experiments: one with random values, and one based on the real case of the Great Valparaíso Fire in Chile, 2014 [20]. Both sets consider the following, in order to simplify the instances to solve the problem:

- The demand on each collection point is a multiple of the bus capacity [13].
- The capacity of each shelter is a multiple of the bus capacity [13].
- Collection point nodes are not connected to each other.
- Shelter nodes are not connected to each other.
- All distances in the graph are symmetric, i.e. $d_{ij} = d_{ji}$ [2].
- Total shelter capacity is always greater or equal to the total demand on collection points.
- Buses are evenly distributed between all yards.

Dataset 1: The Great Fire of Valparaíso: The Great Fire of Valparaíso is a wildfire that took place in Chile in 2014. It started as a forest fire and then spread to an inhabited area, consuming a total of 1,000ha, and burning close to 2,900 homes according to local sources [20]. It displaced approximately 12,500 people, from which around 2,500 were taken to shelters. This dataset considers the main aspects of the event, in terms of number of evacuees, affected area and used shelters, among other aspects. In order solve the real scenario using the proposed BEP model, the following assumptions are considered:

- The shelters and their capacities were estimated according to local reports of total sheltered people. 12 shelters with a total capacity of 2,250 were considered.
- The total number of evacuees is set equal to the total shelter capacity, i.e. 2,250 people, distributed among the collection points considering the population density of the area.
- For the collection points, we used bus stops and open spaces inside the risk zone, to ensure that the buses were able to reach the evacuees. Due to the geography of the zone, 52 collection points were considered.
- Yards are the public transport stations near the endangered area, 5 in total.
- The public transport of the city has been considered, which are buses with capacity for 30 passengers. The number of buses considered goes from 10 to 75 buses, evenly distributed among all yards.
- The distances between nodes were calculated using the Google Maps Distance Matrix API, and are assumed symmetrical.

Figure 3 shows the proposed scenario.

Dataset 2: Random Instances: In total, we generated 9 sets of increasing size, with 10 instances each. Based on [13], the bus capacity in all instances is 1, arc distances and shelter capacities are in the range {1,...,10}, and collection point demands are in {1,...,5}. All generated instances are feasible. Table I shows the size of the instances, where Y is the number of yards, P the number of collection points, S the number of shelters and B the number of available buses.

B. Experimental Setup

1) Experiments on Dataset 1: For the real based scenario, we first ran the algorithm to test different parameter values. We

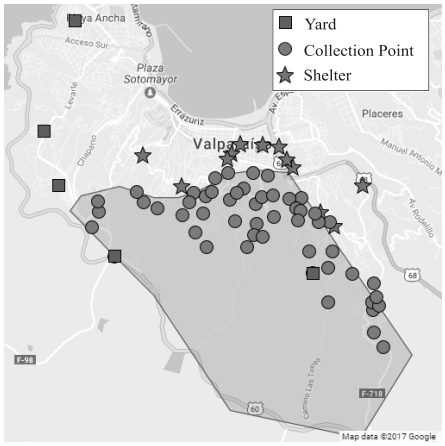

Fig. 3. Evacuation scenario for the Great Fire of Valparaíso (image copyright 2017 Google)

Set	Y	P	S	B
S_2	1	2	2	2
S_3	1	3	3	2
S_4	1	4	4	3
S_5	1	5	5	3
S_6	1	6	6	4
S_7	1	7	7	4
S_8	1	8	8	5
S_9	1	9	9	5
S_{10}	1	10	10	6

TABLE I. RANDOMLY GENERATED INSTANCES DETAIL.

tested parameter α in $\{0.1, 0.3, 0.5\}$, and the maximum HC iterations in $\{20, 50, 100, 500\}$. The execution time was tested with 180 and 300 seconds (3 and 5 minutes respectively), considering [12]. Eight seeds were used, making a total of 192 executions. To corroborate the resulting best parameter values, we used the automatic parameter tuner ParamILS [16]. Finally, we tested the performance of the evacuation in relation with the number of buses. This analysis is inspired on [2], where the performance of the min-max function is evaluated according to the number of vehicles, in order to determine an optimal quantity, since the use of these vehicles has a cost.

Since the instance information is in meters, to estimate the evacuation duration we consider the buses travel at a speed of $40kmph$. This value was chosen considering that the speed limit on an urban area is $60kmph$, and that this evacuation scenario is performed on hills.

2) Experiments on Dataset 2: For the randomly generated instances, we perform a parameter analysis as well. We tested parameter α in $\{0.1, 0.3, 0{,}5\}$, and the maximum number of iterations for the hill climbing in $\{20, 50, 100\}$. Each execution was set to 30 seconds, using four different seeds. Once selected the best parameter values, we corroborated these results with ParamILS [16], and then tested the performance of the algorithm with 15 minutes of execution, using the same seeds, and comparing the best results with CPLEX performance on the same time. When CPLEX did not find the optimal solution on time, the best solution found is given. The algorithm is only compared against MIP, since no algorithm from the literature was available to perform comparisons.

3) Experimental Environment: The experiments were executed in a machine with an Intel core i3 2.53 GHz processor, 6 GB RAM and Ubuntu 16.04 LTS. For the randomly generated instances, we used the MIP solver CPLEX v. 12.6.

RCL size proportion α	Avg distance [m]	Standar dev. [m]	Best result [m]
0.1	12,407	215.36	12,086
0.3	12,286	338.83	11,873
0.5	12,347	386.75	11,879

TABLE II. RESULTS FOR DIFFERENT RCL SIZES ON DATASET 1.

Max. HC It.	Avg distance [m]	Standard dev. [m]	Best result[m]
20	12,799	250.60	12,305
50	12,221	156.70	11,873
100	12,185	170.95	11,873
500	12,181	168.77	11,873

TABLE III. RESULTS FOR DIFFERENT HC MAXIMUM ITERATIONS ON DATASET 1.

C. Results

1) Results for Dataset 1:

Parameter Tuning: The parameter tuning was executed on an instance with 5 yards, 52 collection points, 12 shelters and 50 buses. In Tables II, III and IV, results are shown in terms of the objective function, which is calculated in meter units. In this case, the best value for the RCL size parameter is 0.3, resulting in lower values for the objective function. This value gives a good balance for the creation of a good quality initial solution, while adding diversity to explore different areas of the search space.

For the HC parameter, we notice that 500 iterations give a better average, but the best value obtained is the same with 50 and 100 iterations. This value can be a local optimum solution, or the best solution the implemented move can find, so HC does not necessarily complete all the iterations, and stops earlier in a local optimum. Finally, for the execution time we have that increasing the time gives a better average for the objective function value, but does not improve the best solution found.

Number of buses used in evacuation: Using the parameters selected from the previous analysis, a RCL size of 0.3, maximum HC iterations of 100 and an execution time of 180 seconds, we now evaluate the change on the evacuation duration according to the number of buses used. Table V and Figure 4 show the average evacuation times obtained for each instance, using four different seeds and considering a speed of $40kmph$. Results show that the evacuation time does not decrease linearly with the amount of buses, and has its better value with a fleet size near to 60. Like stablished in [2], incorporating more buses to the evacuation does not necessarily improve the result, but worsens it after this threshold.

In general, we notice that using 10 buses results in an evacuation time of 54 minutes, while with 60 buses the evacuation only lasts 18 minutes. The available time to perform the evacuation will always depend on the situation, and the people in charge of the planification. Since the wildfire was active during four days, spreading through the wildland to the urban area, and the evacuations were performed gradually, an evacuation of 54 minutes for the whole zone seems a good approximation.

2) Results for Dataset 2:

Time limit [s]	Avg distance [m]	Standard dev. [m]	Best result[m]
180	12,382	331.54	11,873
300	12,312	314.00	11,873

TABLE IV. RESULTS FOR DIFFERENT TIME LIMITS ON DATASET 1.

Buses	Avg distance [m]	Avg duration [min]
10	35730.50	53.6
20	19371.50	29.1
30	15092.25	22.6
40	13008.50	19.5
50	12157.25	18.2
60	11992.50	18.0
70	13463.00	20.2
75	18534.00	27.8

TABLE V. OBJECTIVE VALUE AND EVACUATION TIME FOR DIFFERENT NUMBER OF BUSES.

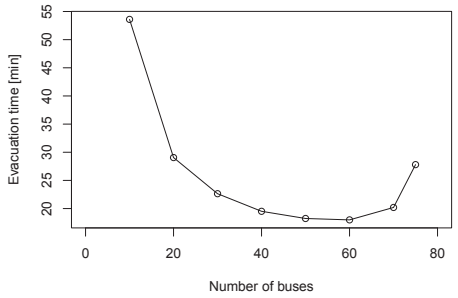

Fig. 4. Time of evacuation for different number of buses.

	RCL size proportion α			Max. HC It.		
	0.1	0.3	0.5	20	50	100
S_2	**1.000**	**1.000**	**1.000**	1.000	1.000	1.000
S_3	1.046	1.046	**1.000**	1.031	1.031	1.031
S_4	1.161	1.020	**1.005**	1.062	1.062	1.062
S_5	1.090	**1.006**	1.028	1.041	1.041	1.041
S_6	1.167	**1.011**	1.076	1.085	1.085	1.085
S_7	1.114	**1.021**	1.124	1.086	1.086	1.086
S_8	1.051	**1.030**	1.129	1.070	1.070	1.070
S_9	1.062	**1.047**	1.212	1.107	1.107	1.107
S_{10}	**1.003**	1.077	1.257	1.112	1.112	1.112
Mean Rank	2.170	1.590	2.240	-	-	-

TABLE VI. AVERAGE RESULTS FOR DIFFERENT PARAMETER VALUES ON DATASET 2.

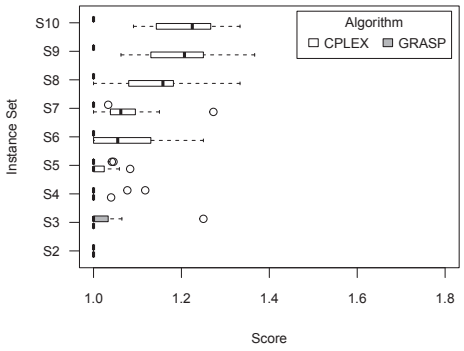

Fig. 5. Boxplot of scores per instance set for CPLEX and GRASP with the best α for each set.

Parameter Tuning: To compare the results between instances of the same set, we stablished a normalization method. A normalized score is calculated as T_i/T_{best}, where T_i is the resulting value for an execution of an instance with a specific set of parameters, and T_{best} is the value of the best solution found for the same instance with any set of parameters. Table VI shows the results for Dataset 2, where each cell is the mean result for an instance set with a specific parameter value. For the RCL size, we can observe that the best value depends on the instance size. The smaller instances, S_3 and S_4, find better results with a bigger proportion of the total candidate list. A size of 0.3 works better for instances from sets S_5 to S_9, and, finally, set S_{10} works better with the smallest size, 0.1. Set S_2 is optimally solved with any value. The feasible trip list size is an important factor behind these results. As the instance size gets bigger, the trip list goes from being too short, to too big, so the value of α has to be big for smaller instances, and small for the bigger ones. Adapting this parameter, the RCL size has an adequate balance to make a non-deterministic greedy construction, accepting a certain degree of randomness. The mean ranks for Friedman test are shown on the last row of Table VI. With a 95% of confidence, we see that α equals 0.3 is the most adequate for Dataset 2.

For the maximum number of HC iterations, the values tested do not change the performance, meaning that the local search converges to a local optimum with less than 20 iterations on most situations, so there is no need to perform more iterations to improve the solution. This allows algorithm to perform more GRASP iterations, exploring more with new constructed solutions.

Performance Comparison: Using the parameter values resulting from the previous experiments, we now compare our best results with the ones from the MIP solver. Here, we use a different normalization method to compare results globally. This is calculated as $T_{alg}/min(T_{grasp}, T_{cplex})$, i.e. the result of an algorithm is normalized by dividing it by the best result obtained for that instance by any of the algorithms. For our approach, we used two different parameter settings: one with α equals 0.3 for all the dataset, and one with the best value of α for each set (shown in Table VI). Results are shown in Figures 5 and 6. For the smallest instance set, both algorithms perform equally well. Using the best α for each set, we have that on S_3 and S_4 the performance is almost the same, except for a few instances where CPLEX obtain slightly better results, although only a few are known optimal. From S_5 to S_{10}, our algorithm is able to find better results, having that CPLEX produces worst values as the instance size grows. On bigger instances, CPLEX has difficulties on producing results, consuming lots of memory resources and using the whole time in finding a solution. On Figure 6, results are very similar. Only sets S_3 and S_{10} where affected, with higher scores for GRASP on S_3, and lower scores for CPLEX on S_{10}.

Table VII shows the results on the Wilcoxon test, comparing the performance of CPLEX with our approach, using α equals 0.3. With a p-value statistic of 0.00 (Table VIII), we see that there is a significant difference on the mean performance of each algorithm. The ranks show that our approach performs statistically better than the MIP solver.

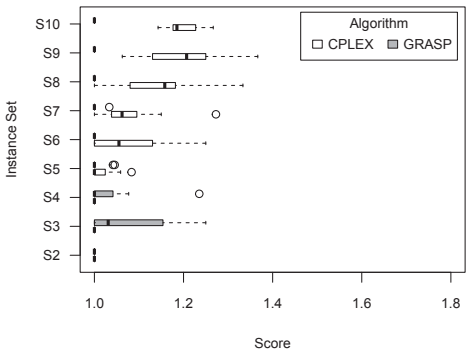

Fig. 6. Boxplot of scores per instance set for CPLEX and GRASP with $\alpha = 0.3$.

	CPLEX - GRASP	N	Mean Rank	Sum of Ranks
$a.\, cplex < grasp$	Negative Ranks	13^a	20.35	264.50
$b.\, cplex > grasp$	Positive Ranks	46^b	32.73	1505.50
$c.\, cplex = grasp$	Ties	30^c		
	Total	89		

TABLE VII. RANKS ON WILCOXON SIGNED-RANK TEST FOR ALL SETS.

		$cplex - grasp$
a. Based on negative ranks.	Z	-4.684^a
b. Wilcoxon Signed Ranks Test.	Asymp. Sig. (2-tailed)	.000

TABLE VIII. TEST STATISTICS.^{b}

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we solved the Bus Evacuation Problem, using a model based on [2] and [13]. For this we presented an algorithm proposal based on the Greedy Randomized Adaptive Search Procedure metaheuristic, which consists of an iterative repetition of a construction phase, and a local search to improve the generated solution. To test the approach, we proposed two sets of instances: the first one is based on a real-world scenario of a recent wildfire occurring in Valparaíso, Chile; the second dataset was built with random values, based on the instances proposed on [13]. The proposed algorithm was able to find reasonable evacuation times for the studied scenario. The best result was 11,992 meters, which corresponds to an evacuation duration of 18 minutes at $40kmph$, considering only the trips, without the time for getting on and off the bus. This result was obtained with only 3 minutes of execution of the algorithm, and with a low usage of memory, which could not be possible to do solving the MIP formulation with CPLEX. The implemented approach is also able to find different solutions for a same objective function value, giving more alternatives to the person in charge of planning the evacuation.

As future work, we plan to solve more complex problem instances considering, for example, cycles in graphs and asymmetric costs, to add different paths between collection points and shelters. Additionally, it may be interesting to apply some strategies to this approach, like detecting pairs of nodes that improve the performance when visited consecutively, or completely satisfy the demand in a collection point before tending to another, to study the performance on these cases. We would also like to improve our approach considering more moves for the local search, exploring other zones on the search space. Finally, an emergency scenario like a wildfire can benefit from a dynamic version of the problem, reorganizing the location of collection points and shelters as the fire spreads through the zone.

REFERENCES

[1] H. Abdelgawad, B. Abdulhai, and M. Wahba. Multiobjective optimization for multimodal evacuation. *Transportation Research Record: Journal of the Transportation Research Board*, (2196):21–33, 2010.

[2] D. Bish. Planning for a bus-based evacuation. *OR spectrum*, 33(3):629–654, 2011.

[3] B. Crevier, J.F. Cordeau, and G. Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773, 2007.

[4] T. Dhamala. A survey on models and algorithms for discrete evacuation planning network problems. *Journal of Industrial and Management Optimization*, 11(1):265–289, 2015.

[5] J. Elliott and J. Pais. Race, class, and Hurricane Katrina: Social differences in human responses to disaster. *Social Science Research*, 35(2):295–321, 2006.

[6] T. Feo and M. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71, 1989.

[7] T. Feo and M. Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[8] P. Festa and M. Resende. Grasp: An annotated bibliography. In *Essays and surveys in metaheuristics*, pages 325–367. Springer, 2002.

[9] P. Festa and M. Resende. An annotated bibliography of grasp–part i: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009.

[10] P. Festa and M. Resende. An annotated bibliography of grasp–part ii: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.

[11] M. Goerigk, K. Deghdak, and P. Heßler. A comprehensive evacuation planning model and genetic solution algorithm. *Transportation research part E: logistics and transportation review*, 71:82–97, 2014.

[12] M. Goerigk and B. Grün. The robust bus evacuation problem. Technical report, Fachbereich Mathematik, Technical University of Kaiserslautern, 2012.

[13] M. Goerigk, B. Grün, and P. Heßler. Branch and bound algorithms for the bus evacuation problem. *Computers & Operations Research*, 40(12):3010–3020, 2013.

[14] M. Goerigk, B. Grün, and P. Heßler. A branch-cut-and-price approach to the bus evacuation problem with integrated collection point and shelter decisions. Technical report, Technische Universität Kaiserslautern, Fachbereich Mathematik, 2013.

[15] S. He, L. Zhang, R. Song, Y. Wen, and D. Wu. Optimal transit routing problem for emergency evacuations. In *Transportation Research Board 88th Annual Meeting*, number 09-0931, 2009.

[16] F. Hutter, H. Hoos, K. Leyton-Brown, and T. Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009.

[17] P. Murray-Tuite and B. Wolshon. Evacuation transportation modeling: An overview of research, development, and practice. *Transportation Research Part C: Emerging Technologies*, 27:25–45, 2013.

[18] L. Özdamar and M. Ertem. Models, solutions and enabling technologies in humanitarian logistics. *European Journal of Operational Research*, 244(1):55–65, 2015.

[19] U. Pyakurel, M. Goerigk, T. Dhamala, and H. Hamacher. *Transit dependent evacuation planning for Kathmandu valley: A case study*. Technische Universität Kaiserslautern, Fachbereich Mathematik, 2014.

[20] P. Reszka and A. Fuentes. The great valparaiso fire and fire safety management in Chile. *Fire Technology*, 51(4):753–758, 2015.