Enumeration of Pareto Optima for a Bicriteria Evacuation Scheduling Problem

Kaouthar Deghdak, Vincent T’kindt and Jean-Louis Bouquard
Université François-Rabelais Tours, Laboratoire d’Informatique (EA 6300)
Equipe Ordonnancement et Conduite (ERL CNRS 6305), 64 Avenue Jean Portalis, 37200 Tours, France

Keywords: Scheduling Evacuation, Branch and Bound, Greedy Heuristic, Matheuristic.

Abstract: In this paper, we consider a large-scale evacuation problem after an important disaster. We model the evacuation of a region from a set of collection points to a set of capacitated shelters with the help of buses, thus leading to scheduling the evacuation operations by buses (Bus Evacuation Problem, BEP). The goal is twofold; first, minimizing the total evacuation time needed to bring the resident out of the endangered region, and secondly, minimizing the total exposure to danger. The resulting problem is a bicriteria problem. We propose a time-indexed formulation, as well as several approaches for finding both upper and lower bounds for BEP used within a branch and bound algorithm. In computational experiments, we analyse and evaluate the efficiency of the proposed solution algorithms.

1 INTRODUCTION

After a natural disaster the evacuation of people from endangered region becomes necessary. Evacuating of urban area is a highly difficult and complicated task that requires the efficient utilization of the transportation network to facilitate the movement of evacuees to safety. The major issue is congestion, which can cause an extremely dangerous situation and life threatening. Therefore, the choice of routes on which people are evacuated is a crucial aspect, which leads to the success or not, of an evacuation plan. Several papers tackle the routing evacuation problem; to model these problems, network flow approaches or traffic assignment approaches are used. Early models focus on building evacuation, as for example (Chalmet et al., 1982), (Choi et al., 1988) and (Mamada et al., 2005). We refer the reader to the survey of (Hamacher and Tjandra, 2001) that discusses evacuation models and methods for building evacuation.

For urban area evacuation, Yamada (Yamada, 1996) models city evacuation as a minimum cost flow problem to assign the pedestrian evacuees to shelters, knowing that the routes are calculated by solving shortest path problems. (Lu et al., 2005) use a static network with time dependent capacity on node as well as time dependent arc capacity to define an evacuation plan i.e., define routes and timetables to minimize the total evacuation time. To solve this problem an heuristic is presented, later on improved after by Kim et al. (Kim et al., 2007) in terms of running time. Lim et al. (Lim et al., 2009) propose an evacuation plan for an urban area in case of an hurricane evacuation. They use a maximum dynamic flow approach to find the maximum number of evacuees transported outside the damaged area within a given time period. A variety of dynamic network flow problems are considered for region evacuation in Bretschneider (Bretschneider, 2012). The decision variables in these models are the evacuees’ flow, the number of lanes in each arc of the network and the traffic routing. This study focuses on the routing within intersection nodes and assures that no crossing conflict occurs. All these models are solved heuristically.

Dynamic traffic assignment modelling has been extensively applied to regional vehicle evacuation problems. The dynamic traffic assignment problems allow to model time varying flow, different traffic status and phenomena, like congestion. (Sattayhatewa and Ran, 2000) are the first who suggest the use of the system optimum traffic assignment model in the evacuation context to minimize either the total evacuation time or the travel time between each origin-destination pair. However, the model has been only tested in a three-link network. Han et al. (Han et al., 2006) analyse a different problem which is modelled as a simple system optimum traffic assignment problem like the one proposed by Sheffi et al. (Sheffi et al., 1982). This model allows multiple sources (collection points) but only a single destination (shelter) and the
objective is to minimize the total travel cost for all evacuees in the network. The cell transmission model (CTM), introduced by Daganzo (Daganzo, 1994), has recently obtained significant attention of evacuation modellers. The basic idea of the CTM is to divide the network into homogenous cells that can be traversed by a vehicle in one time period in free flow traffic. The first one who uses this model is Ziliaskopoulos (Ziliaskopoulos, 2000). He proposes a linear model for a system optimum dynamic traffic assignment problem with a single sink. Liu et al. (Liu et al., 2006) model a large scale evacuation problem using the model proposed by (Ziliaskopoulos, 2000). They propose two evacuation models: in the first model, the goal is to maximize the number of evacuees reaching the destination within a given time horizon. The second one aims at minimizing the total evacuation time. Other similar studies that use cell transmission models include the work of Petta and Ziliaskopoulos (Petta and Ziliaskopoulos, 2001), and Chiu et al (Chiu et al., 2007). Interesting studies look at more realistic situations that capture the uncertain factors of the risk of a disaster. Yazici and Ozbay (Yazici and Ozbay, 2007) consider uncertain roads capacities while Ng and Waller (Ng and Waller, 2010) take into consideration the uncertain number of evacuees. These robust evacuation models are also based on cell transmission models.

Recently, Bish (Bish, 2011) has introduced and studied a new model for bus-based evacuation planning. The choice of buses as a transportation mode is motivated by the fact that car-based evacuation is logistically complex, expensive, produces unacceptable levels of congestion, and is more dangerous than bus-based evacuation. To solve the bus evacuation problem, Bish proposes a mixed integer program and two heuristics. Goerigk et al. (Goerigk et al., 2013b) consider a problem closely related to the one discussed in (Bish, 2011), for which they propose several Branch-and-Bound algorithms. Robust bus evacuation models have been considered in (Goerigk and Gruen, 2014) and (Goerigk et al., 2013a).

A recent study is the one conducted by Bretschneider (Bretschneider, 2012) in which he introduces the multiple commodity evacuation problem using buses and vehicles. The author proposes a mixed integer program, where the number of lanes in each arc is represented by integer variables. The lanes are partitioned into public and emergency lanes but only within intersection. The objective function of the proposed model is to minimize a weighted linear combination of the flows of the commodities arriving at their corresponding destinations and the total number of emergency lanes used. This problem is solved heuristically and the proposed heuristic is only able to solve small instances in a reasonable amount of time.

The problem addressed in this paper is related to the one discussed in the paper of Bish (Bish, 2011). We consider the evacuation of people due to a natural disaster such as an earthquakes, where evacuees have to change their centre of lives from several days to several months with the eventual goal of returning to their respective homes. In particular, we assume that the locations of shelter (i.e., locations to which people are evacuated, outside the damaged area), the locations of collection points (i.e., where people are gathered waiting to be evacuated) and the capacitated transportation network are known. The goal is to define a macroscopic plan of evacuation, implying people are considered homogeneously, i.e., the evacuees are assumed to share the same behaviour and must be transported from the collection points to the shelters. During the evacuation it is efficient to use roads that pass through safe area and not through the endangered zone. The evacuees aim is to reach a shelter without being injured as fast as possible. The evacuation is performed by a set of homogeneous buses. In contrast of the works of Bish (Bish, 2011) and Goerigk et al. (Goerigk et al., 2013b), which minimize the maximum travel time over all buses. We deal with a bicriteria problem, where the total evacuation time and the risk exposure of the evacuees are minimized. The remainder of the paper is organized as follows. In Section 2, we describe the Bus Evacuation Problem in details. In Section 3, we provide a mixed-integer programming formulation. In Section 4, we present Branch and Bound method, and provide lower bounds, an upper bound and we discuss branching rule. Computational results are presented in Section 5. Finally, Section 6 concludes the paper.

2 PROBLEM STATEMENT

Consider a network \((\mathcal{N}, A)\), where \(\mathcal{N}\) and \(A\) denote the set of nodes and edges, respectively. \(\mathcal{N}\) is composed of two subsets of nodes: \(\mathcal{P} = \{1, \ldots, P\}\) and \(\mathcal{S} = \{1, \ldots, S\}\). \(\mathcal{P}\) is a set of collection points where evacuees are initially located, and \(\mathcal{S}\) is a set of shelters. An edge \((k, j) \in A\) exists, iff evacuees can be transported from collection point \(k\) to shelter \(j\). A set \(\mathcal{B} = \{1, \ldots, B\}\) of identical buses is used to evacuate people. The number of evacuees at every collection point \(k\) is known, and given in terms of integer multiples of bus loads, denoted by \(d_k\). Furthermore, we denote by \(\mathcal{M} = \{1, \ldots, M\}\) the set of evacuation operations, where \(M = \sum_{k \in \mathcal{P}} d_k\). Each operation is also defined by a collection point at which the correspond-
ing people to evacuate are located. Any shelter \( j \in S \) has a capacity \( cap_j \) expressed as a number of buses that can bring evacuees. We denote by \( p_{i,j} \) and \( r_{i,j} \) the traveling time and risk value, respectively, of an evacuation operation \( M_i \) starting at time \( t \) from collection point \( P_i \) toward shelter \( S_j \). The risk values correspond to the likelihood of buildings collapse or the risk of congestion network. Travel times and risk values are time-dependent. This means that they can be increased or decreased, over time, depending on the state of the network. This is a consequence of the evolution of the transportation network through time due to events such as earthquake aftershocks, road repairs, roads congestion, etc. We consider that such \( s \) events happen at known times \( t_s \) and we define:

\[
p_{i,j} = p_0 + \begin{cases} 
    a_{i,j}^0 & \text{if } t \in [0,t_1] \\
    a_{i,j}^1 & \text{if } t \in [t_1,t_2] \\
    \vdots & \\
    a_{i,j}^{s-1} & \text{if } t \in [t_{s-1},t_s] 
\end{cases}
\]

\[
r_{i,j} = \begin{cases} 
    b_{i,j}^0 & \text{if } t \in [0,t_1] \\
    b_{i,j}^1 & \text{if } t \in [t_1,t_2] \\
    \vdots & \\
    b_{i,j}^{s-1} & \text{if } t \in [t_{s-1},t_s] 
\end{cases}
\]

where \( a_{i,j}^s \) and \( b_{i,j}^s \) are the travel times and the risk values, respectively, if a bus starts evacuation operation \( M_i \) from collection point \( P_i \) toward shelter \( S_j \) in the \( s \)th time interval, i.e., it’s starting time \( t \in [t_{s-1},t_s] \). The number of finite intervals \([t_{s-1},t_s]\) is determined by a preliminary forecasting of the evolution of the transportation network. The constant \( p_0 \) is the average travel time between the shelters and the center of the damaged area. This constant is an estimation of the returning time of empty buses. This approach to defining the \( p_{i,j} \)’s enables us to approximate and therefore make simpler the bus routing problem. Throughout the paper, we refer to the scheduling of evacuation operations as the transportation of evacuees from collection points to shelters using buses.

The problem we consider is to find a schedule such that all evacuees are transported from the collection points \( P \) to the shelters \( S \), minimizing the maximum evacuation time denoted by \( T_{\text{evac}} \) and the sum of the risk values denoted by \( R \). It is important to notice that criteria \( T_{\text{evac}} \) and \( R \) are potentially conflicting since the fastest evacuation routes may not be the safest ones, or because those routes which are the fastest and the safest have a limited capacities (thus requiring to use non fastest and safest routes).

In the field of multicriteria optimization, many methods to compute Pareto front are known (T’kindt and Billaut, 2006). In this work we use the \( \varepsilon \)-constraint approach as follows: the total risk \( R \) is minimized under the constraint that the maximum evacuation time \( T_{\text{evac}} \) is lower or equal to given value \( \varepsilon \). By solving this problem for different values of \( \varepsilon \), the set of all pareto optima can be computed. In the first step, \( \varepsilon \) is equal to the upper bound of the objective function \( T_{\text{evac}} \). The solution which is obtained \((T_{\text{evac}}, R)\) is added to the set of solutions. We set \( \varepsilon = T_{\text{evac}} - 1 \) and iterate. If no solution is obtained, then there is no feasible solution and the procedure stops.

From practical point of view, solving one \( \varepsilon \)-constraint problem, as defined above, makes sense: the \( \varepsilon \) value represents a threshold which guarantee that the evacuation is performed in no more than \( \varepsilon \) time units. Then the aim becomes at minimizing the total risk within that time limit. In this paper we are interested in enumerating the set of Pareto optima for criteria \( T_{\text{evac}} \) and \( R \) by iteratively solving \( \varepsilon \)-constraint problems as defined above. Additionally, while the evacuation time is a very descriptive value, the total risk is a more abstract value, and fixing a desired quality is hardly possible in practice. Thus, this implies that the other possible \( \varepsilon \)-constraint problem (minimizing \( T_{\text{evac}} \) under the constraint \( R \leq \varepsilon \)) loses interest.

### 3 Mathematical Programming

To model the Bus Evacuation Problem we have proposed a time-indexed mathematical formulation. This choice is motivated by the paper of (Berghman et al., 2010), in which various mathematical formulations for a parallel-machine scheduling problem are compared. This problem represents a dock assignment problem which is related to BEP. They showed that their time-indexed formulation performs significantly better than other formulations (which were assignment-based and flow-based). The drawback of this formulation is the presence of a pseudo-polynomial number of variables. Let us turn to the model for our evacuation problem in which \( T_{\text{evac}} \) is the desired quality of the criterion \( T_{\text{evac}} \), \( T \) is the time horizon, and \( T = \{1, \ldots, T\} \) is the set of time points. The variables are as follows:

\[
\forall i \in M, \forall j \in S, \forall t \in T; \\
x_{i,j,t} = \begin{cases} 
    1 & \text{if a bus starts evacuation operation} \\
    i \text { towards } j \text { at } [t,t+1[, \\
    0 & \text{otherwise.}
\end{cases}
\]

\( R \): the total risk exposure.
The proposed (IP) formulation for the $\varepsilon$-constraint problem, is as follows:

$$\min R$$  
(1)

Subject to:

$$R \leq \sum_{t \in T} \sum_{i \in M} \sum_{j \in S} x_{i,j,t} \cdot r_{i,j,t}$$  
(2)

$$T_{evac}^{exp} \geq (t + p_{i,j,t})x_{i,j,t}$$  
(3)

$$\forall i \in M, \forall j \in S, \forall t \in T$$  
(4)

$$\sum_{t \in T} x_{i,j,t} \leq cap_j$$  
(5)

$$\forall j \in S$$

$$\sum_{t \in T} \sum_{i \in M} \sum_{j \in S} x_{i,j,t} \leq B$$  
(6)

$$\forall t \in T$$

$$\sum_{i \in M} \sum_{j \in S} x_{i,j,t} = 1$$  
(7)

$$\forall i \in M$$

$$x_{i,j,t} \in \{0, 1\}$$

(8)

where \( Seq[j] \) is the operation in the position \( j \) in the sequence \( Seq \), and \( T_i \) is the set of time points, when the travelling times and the risk values of the operation \( i \) will be change. Notice that the constraint (3), which also holds for the subsequence Seq, can be easily answered by setting to $+\infty$ any value $F(\{i\}, t + p_{i,j,t}) + r_{i,j,t}$ as soon as $t + p_{i,j,t} > T_{evac}$.

The optimal value of the total risk for the sequence Seq of a given bus is then $F(\{\emptyset\}, 0)$. This can be computed in $O(k|Seq|)$ time with $k = \max_{i \in Seq(T_i)}$.

4.2 Lower Bounds

In the following, we present three lower bounds on the sum of the risk values for an instance of BEP. The first one is based on the linear relaxation of the BEP model, the second one is based on a knapsack formulation and the last one is a greedy heuristic. Note that these three lower bounds can be computed in polynomial time.

**LB1.** The first and most intuitive lower bound consists in solving the continuous relaxation of the (IP) model proposed previously. At each node of the search tree, all the variables already scheduled, i.e., the variables of the partial solution are fixed to 0 or 1. Then the relaxed model is solved with the fixed variables.

**LB2.** Since knapsack problems are maximisation problems, we can use an upper bound proposed in the literature to calculate lower bounds for an instance of BEP. We can model a simplified version of BEP as a multiple knapsack problem with the relaxation of the bus constraints (5) as follows:
max \( \sum_{l \in M \times J} \sum_{j \in S} x_{i,j}^l * r_{i,j}^l \) \( \quad (10) \)
\[
\sum_{l \in M \times J} \sum_{j \in S} x_{i,j}^l \leq \text{cap}_j \quad \forall j \in S
\]
\( x_{i,j}^l \in \{0, 1\} \) \( \quad (12) \)

where all data are positive and integer. Each variable \( x_{i,j}^l \) is associated with a variable \( x_{i,j} \) of the (IP) formulation. Similarly, each value \( r_{i,j}^l \) is associated with a risk value \( r_{i,j} \) and is defined by \( r_{i,j}^l = C - r_{i,j} \), with \( C \) a fixed constant such that \( C \gg r_{\text{w,vc}} \), \( u \in M, v \in S, w \in T \). We use the upper bound proposed by (Pisinger, 1995), the following steps are used to calculate the lower bound of BEP:

- At each node and for each bus, we calculate \( T_{\text{evac}}^\text{opt} \) of the subsequence of already scheduled operations. This can be done in polynomial time by iteratively determining the starting time of the operations in the order they appear in the sequence of a given bus. The \( T_{\text{evac}}^\text{opt} \) is calculated to obtain the minimum starting time of the unscheduled operations.
- Update the residual capacity \( \text{cap}_j \) of shelter \( j \) taking account of the scheduled operations.
- All variables \( x_{i,j}^l \) related to unscheduled operations are sorted by non-increasing order of \( r_{i,j}^l \). The first one is then selected and fixed to 1, and all other variables \( x_{i,j} \) related to the same operation than the selected variable are removed from that sorting an fixed to 0. Besides, the residual capacities are updated.

From the obtained values of the \( x_{i,j} \)'s, we can deduce the values of the \( x_{i,j}^l \)'s. Then LB2 is calculated as the sum of the risk for that variables plus the total risk of the scheduled operation in this node using the recursive equations (8) and (9).

\( \text{LB3} \) This bound is calculated in a similar, but even simpler, way than LB2, however, here, for the unscheduled operations for an incumbent node, we compute \( r_i' = \min_{j \in S, i \in T} r_{i,j} \). Then the lower bound on the risk generated by the unscheduled operations is given by \( \Sigma r_i' \).

4.3 Upper Bounds

UB. To construct a feasible solution of BEP, we propose a local search method called matheuristic. The general idea of matheuristics is to exploit the strength of both metaheuristic algorithms and exact methods as well, leading to a hybrid approach (Della Croce et al., 2014).

As any local search method, we need to construct a feasible solution, which will be improved afterward using the matheuristic. For this, we develop two greedy approaches. These approaches are used to enumerate the Pareto front. Figure 1 presents the general algorithm of the proposed greedy heuristic, which takes upon entry a sequencing rule \( R \) and the desired value of the total evacuation time \( T_{\text{evac}}^\text{opt} \). Let \( \text{oprl} \) be the set of the evacuation operations that will be assigned on buses, \( \text{sortlist}(i,j) \) be the set of sorted operations according to the rule \( R \) at a given time \( t \), \( \text{feassortlist}(i,j) \) be the set of operation from \( \text{sortlist}(i,j) \) for which \( \text{LB}_{\text{evac}} < T_{\text{evac}}^\text{exp} \), \( T_{\text{evac}}^\text{exp} \) be the total evacuation time of a bus \( b \), and \( R^B \) the sum of the risk values accumulated when the bus is used. The greedy heuristic's solution is stored in \( \text{scheudelist} \). Each operation added in \( \text{scheudelist} \) will be deleted from \( \text{oprl} \), \( \text{sortlist}(i,j) \) and \( \text{feassortlist}(i,j) \).

The function event, presented in the pseudo code of the greedy algorithm (Figure 1), checks if there is an event requiring to recalculate the priority list \( \text{sortlist} \). Each rule \( R \) has a specific event function. Notice that an external archive, referred to as \text{nondominated}, is maintained through different calls of the greedy heuristic. This archive contains the set of Pareto optima computed so far. Also the greedy heuristics are ran for different values of \( T_{\text{evac}}^\text{exp} \). Each time, for a given \( T_{\text{evac}}^\text{exp} \) value, a greedy heuristic finds a feasible solution \( \text{scheudelist} \), the function UpdateArchive( ) is called to update the current set of Pareto optima. If \( \text{scheudelist} \) is not dominated by a solution from \text{nondominated}, then it is added to that set. If a solution from \text{nondominated} is dominated by \text{scheudelist} then it is removed from the archive.

In the following, we have tested two greedy heuristic versions.

1. Version 1: this version uses rule \( R_1 \). The evacuation operations are sorted according to increasing order of the risk value \( r_{i,j} \).

2. Version 2: this version uses rule \( R_2 \). In this rule, evacuation operations \( \text{oprl} \) are sorted by increasing order of their values \( -\frac{T_{\text{evac}}^\text{exp}}{\sqrt{\Sigma_{t \in T} x_{i,j}^l (r_{i,j} - \bar{r}_{i,j})^2}} \), where the denominator is the standard deviation of the risk value associated to operation \( i \). Using this rule, we would like to ensure that if an operation \( \text{oprl} \) has a small risk value at \( t \) and after this time the risk value for this operation will be huge, then it is preferable to schedule \( \text{oprl} \) at time \( t \).

In these two versions, the function event returns true if there are modifications in the values of evacuation operations’ risk.

Let there be an initial heuristic solution given as
The additional constraints (13) forces the changes to occur within the operation-window. If we have an improvement in \( R^0 \), then, in the new solution \( \tilde{x} \), all of the operations that started after the time \( r+h \) in the initial solution \( x \) will be left time shifted keeping their previous positions and respecting the bus-constraint (5). The idea is sketched in figure 2.

If no improved solution is found, first we test using the function UpdateArchive(), if this solution is nondominated by another solution in the nondominated set. If so, this solution will be added to nondominated set. Second, a new operation-window (i.e., new value of \( r \)) is selected to be optimized, and so on until all possible windows have been selected. The search is stopped if no window reoptimization problem has an optimal solution solution that improves the current solution or if a predefined time limit is exceeded.

**Example 1. Consider the example of Figure 2. We have two shelters, eleven operations and two buses**
available (bus 1 and bus 2). The current solution is depicted in Figure 2(a): operations 3, 8, 4, 10, and 2 are processed by Bus 1 and operations 1, 6, 7, 9, 5, and 11 are processed in Bus 2. The starting time of the operation-window is \( r = 2 \) and the size of this window is \( h = 7 \) (Figure 2(b)). The total evacuation time of this solution is \( T_{\text{evac}} = 17 \), the total risk is \( R = 50 \), the maximum evacuation time of the operation-window is \( T_{\text{evac}}^w = 11 \) and the total risk of the operation-window is \( R = 30 \). The solution obtaining \( \bar{x} \) from the neighbourhood, after the window reoptimization problem has been solved, is given in Figure 2(c). The figure shows that both the total risk, the maximum evacuation time of the operation-window, the maximum evacuation time and the total risk values of the whole schedule have been reduced. Operations 5, 10, 2, and 11 have been left time shifted keeping their previous positions. The total risk values and the maximum evacuation time of the new solution are \( T_{\text{evac}} = 15 \) and \( R = 43 \), respectively.

The algorithm of the matheuristic is given in Figure 3.

### 5 EXPERIMENT RESULT

In this section, we focus on the experimental evaluations of the (IP) solution, the heuristic algorithms, and on the branch and bound algorithm. We first describe how the experimentation has been configured.

**Environment.** All experiments were run on a computer with a 4-core Intel processor, running at 2.60 GHz with 20MB cache, 8 GB RAM and Windows 7. We wrote our code in C++, and used the commercial IP solver CPLEX v.12.6. CPLEX was pinned to one core for the solution of the time-indexed formulation in order to be consistent with the branch and bound algorithm.

**Dataset.** This work is partly a research project, called DSS-Evac-Logistic, and was partially granted by the French research agency ANR. In this project, we consider the real-world instance of Nice (France), as a case study. Therefore, the datasets are randomly generated in such a way that we always have feasible and realistic instances for the city of Nice. The number of shelter \( S \) takes values \( \{2, 4, 6, 8, 10\} \) and the capacity \( \text{cap}_j \) of each shelter draws at random from \( \{20, 21, 22, ..., 40\} \). The number of collection points \( P \) takes values \( \{10, 20, 30, 40\} \) and the number of evacuees \( d_i \) waiting on the collection point \( i \) is generated randomly from the following:

\[
\begin{bmatrix}
1 & 0.9 \sum_j \text{cap}_j \\
\frac{7}{4} & \frac{9}{4} \sum_j \text{cap}_j
\end{bmatrix}
\]

This generation ensures that the total evacuees’ number \( M \) is fewer than the total shelters’ capacity. We assume that the time period of evacuation is \([D, D+1]\), where \( D \) is the day of disaster. In the following, we set time discretization as one-quarter hour, which implies that \( T \) is equal to 192 quarters.

We assume that we have five events happening after time \( 0 \) and changing the value of the operations’ traveling times and risks. To do so, we generated two degradation dates: \( T^{D_1}_1, T^{D_2}_2 \in [0, 192] \), common to all evacuation operations, and three improvement dates: \( T^{I_1}_1 \in U[0, T^{D_1}_1], T^{I_2}_2 \in U[T^{D_2}_2, T^{D_2}_1], T^{I_3}_3 \in U[T^{D_1}_1, T^{D_2}_2] \), specific to each evacuation operation. Operations’ travelling times are drawn randomly from \( \{1, 2, 3, 4, ..., 8\} \) and operations’ risk values are generated randomly from \([0, 10]\), where the value 0 means that roads are completely safe, whilst the value 10 means that roads are very dangerous. Finally, the number of buses \( B = \frac{\sum_{i} d_i + P}{100} \), where \( \bar{p} \) is the average traveling time, which ensures that we have enough buses to accomplish the evacuation in 180 quarters of an hour.

We generate 20 instances for each couple \((P, S)\). For each instance, we run the following algorithms:
• Exact Solution Algorithms: we have tested the solution of the (IP) model by CPLEX solver. In the remainder of the paper, this solution algorithm will be denoted by IP. We also refer to B&B as the solutions of the Branch and Bound algorithm using the LB1, LB2 and LB3, simultaneously as follows: in the top of the three it is clear that LB1 outperform LB1 and LB2, for this reason in the first level of the search tree we calculate in each node only the LB1. For the remainder of level tree, in each node we compute first the LB3, if we cannot prune this node we calculate then the LB2, this may be lead to cut this node.

Notice that we impose a time limit of 2000 seconds and a memory limit of 1 GB per instance and algorithm.

• Heuristic Algorithm:

First, notice that the average values of $Q_3(\text{HAS})$ are very small then the average values of $Q_3(\text{MAS})$ especially when the number of shelter equal to two, which means that matheuristic improve the quality of the solutions obtained by the greedy heuristic. On the other hand, according to the metric $Q_4$, we observe that for all instances, matheuristic helps to spread the points in the objectives space. Moreover, matheuristic allows to approach the approximate set obtained by the greedy heuristic to the reference set delivered by the IP. Notice that the average values of the metrics $Q_3$ and $Q_4$ is always equal to 0.

Table 1 presents the average values of metrics $Q_3$ and $Q_4$. The reference sets are obtained by CPLEX and the approximation sets are obtained by the matheuristic or greedy heuristics. Column #oprs presents the number of evacuation operations, column #inst reports the number of shelters, column #bus presents the number of buses, and column #inst_tot presents the number of instances for each problem size. Notice that the instances are randomly generated based on a number of collection points and a number of shelters. However, the size of the instances solved by CPLEX, B&B and Matheuristic depends on the total number of evacuation operations, i.e., $\sum d_i$. Consequently, after having generating instances we gathered them in classes of "equivalent size instances" but in term of number of evacuation operations, number of shelters and number of buses.

First, notice that the average values of $Q_3(\text{MAS})$ are very small then the average values of $Q_3(\text{HAS})$ especially when the number of shelter equal to two, which means that matheuristic improve the quality of the solutions obtained by the greedy heuristic. On the other hand, according to the metric $Q_4$, we observe that for all instances, matheuristic helps to spread the points in the objectives space. Moreover, matheuristic allows to approach the approximate set obtained by the greedy heuristic to the reference set delivered by the IP. Notice that the average values of the metrics $Q_3$ and $Q_4$ is always equal to 0.

Table 2 presents the running times and the number of explored nodes of the IP and B&B. Columns IP (time(s)) report the average and the maximum running times of CPLEX when enumerating the non-dominated set. Similarly, B&B columns report the
Table 2: Comparison of running times and number of explored nodes.

<table>
<thead>
<tr>
<th>#opers</th>
<th>#shel</th>
<th>#inst</th>
<th># solved_opt</th>
<th>IP (time(s))</th>
<th>B&amp;B (time(s))</th>
<th>#nodes (IP)</th>
<th>#nodes (B&amp;B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>avg</td>
<td>max</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>[40,60]</td>
<td>4</td>
<td>[2,3]</td>
<td>6 0 0</td>
<td>545 560 2000 2000</td>
<td>0 0 45.17</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>[60,70]</td>
<td>4</td>
<td>3</td>
<td>13 0 0</td>
<td>522.3 2000 2000 2000</td>
<td>0 0 27.77</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>[70,80]</td>
<td>4</td>
<td>[3,4]</td>
<td>17 0 0</td>
<td>590.5 2000 2000 2000</td>
<td>0 0 19.65</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>[80,90]</td>
<td>4</td>
<td>4</td>
<td>26 0 0</td>
<td>599.5 2000 2000 2000</td>
<td>0 0 16.58</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>[90,110]</td>
<td>4</td>
<td>[4,5]</td>
<td>18 0 0</td>
<td>636 2000 2000 2000</td>
<td>0 0 12.86</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

CPU times for B&B enumeration approach. Column #solved_opt presents the number of instances that have been solved to optimality by IP and B&B. Additionally, the number of explored nodes is also reported. As the results in Table 2 illustrate, when the number of shelter equals 2, both IP and B&B spend the time limit to enumerate the nondominated set. Furthermore, for some instances, IP is very effective; it is able to enumerate the exact nondominated set for 12 instances, while the B&B to enumerate the exact nondominated set for two instances. In addition, the IP explores fewer nodes than the B&B algorithm. Unfortunately, when the number of operations is larger than 40, in some cases, IP fails to produce feasible solutions due to the imposed memory limit, hence, the number of explored nodes is always equal to 0 and the CPU times is less than 2000 seconds. We can also observe that the number of explored nodes by the B&B are significantly decreased, it is interpreted by the fact that the continuous relaxation of the (IP) using to calculate \( LB_1 \) is very slow.

From Table 2, we can conclude that the IP outperformed the B&B in some instance when the number of shelter equal to 2. But the IP fails to give feasible solution due to memory limit for more larger instance. We can also conclude that the B&B will be competitive if we use two new lower bound more effective than the three lower bounds proposed in this paper.

6 CONCLUSION

In this work, we have studied the Bicriteria problem of scheduling evacuation operations which we have called the bus evacuation problem (BEP). To enumerate the exact nondominated sets, we have provided a time-indexed formulation (IP) for this problem and B&B algorithm. Computational experiment shows that for most instances, neither the IP nor the B&B can enumerate the exact nondominated sets. Next, we have provided two heuristics, the first one is a set of greedy heuristics. The second one is a local search called matheuristic based on the mathematical formulation provided, which improves the greedy heuristic solutions.

Future investigation needs to be done in order to implement in a real-life situation the proposed heuristic algorithms. They will be used in an off-crisis context as a tool for designing evacuation plan. A key issue is now to capture the preferences of the end user in order to select, from the approximated set of Pareto optima, the solution to implement.

ACKNOWLEDGEMENTS

This research has been supported by ANR-11-SECU-002-01, project DSS_EVAC_LOGISTIQUE (CSOSG 2011).

REFERENCES


Daganzo, C. F. (1994). The cell transmission model: a dynamic representation of highway traffic consistent
with the hydrodynamic theory. Transportation Research Part B, 28(4):269.


