The Robust Bus Evacuation Problem *

Should I stay or should I go now?

Marc Goerigk Bob Grün

December 18, 2012

Abstract

By natural or man-made disasters, the evacuation of a whole region or city may become necessary. Apart from private traffic, the evacuation from collection points to secure shelters outside the endangered region will be realized by a bus fleet made available by emergency relief. The arising *Bus Evacuation Problem* (BEP) is a vehicle scheduling problem, in which a given number of evacuees needs to be transported from a set of collection points to a set of capacitated shelters, minimizing the total evacuation time, i.e., the time needed until the last person is brought to safety.

In this paper we consider an extended version of the BEP, the *Robust Bus Evacuation Problem* (RBEP), in which the exact numbers of evacuees are not known, but may stem from a set of probable scenarios. However, after a given reckoning time, this uncertainty is eliminated and planners are given exact figures. The problem is to decide for each bus, if it is better to send it right away – using uncertain numbers of evacuees – or to wait until the numbers become known.

We present a mixed-integer linear programming formulation for the RBEP and discuss solution approaches; in particular, we present a tabu search framework for finding heuristic solutions of acceptable quality within short computation time. In computational experiments using both randomly generated instances and the real-world scenario of evacuating the city of Kaiserslautern, we compare our solution approaches.

1 Introduction

In this paper we consider the problem of evacuating transit-dependent people with the help of buses under uncertainty.

When circumstances like an imminent hurricane or flooding necessitate the evacuation of whole regions or urban areas, operations research is able to play its part in making best use of the available resources to help the affected people in the best possible way. For a survey on problems and methods in evacuation planning, see e.g. [7].

In recent works [9, 4] it is pointed out that the planning process should not only focus on private car-based evacuation, which usually accounts for the

*Partially supported by the Federal Ministry of Education and Research Germany, grant DSS_Evac_Logistic, FKZ 13N12229.

largest part of evacuation traffic, but also on transit-dependent people, like the sick or the elder, who will be transported from the endangered area to provided shelters with the help of buses.

Other authors point out the uncertainty in the problem data that typically occurs in emergency situations, and make use of robust optimization paradigms to find solutions that still perform well when the problem input is disturbed, e.g., [13, 1], who consider an uncertain cell transmission model.

Robust optimization traces its origins to the work of Soyster [10] who considered generalized convex programs. Contrary to the setting of stochastic optimization, it is generally assumed that no probability distribution over the set of possible scenarios is known, resulting in a parameterized family of optimization problems. How to reformulate this family to a single optimization problem, the so-called *robust counterpart*, depends on the application in mind. Plenty possibilities are proposed: As examples, we note the rather conservative approach of feasibility in all scenarios, optimizing the worst-case performance of [3], and two-stage models that give the planner the opportunity to adapt the solution when the scenario becomes known, see [2, 8, 6].

As evacuation planning is typically arranged under tight time constraints, some authors propose meta-heuristics like tabu search [11, 12] to find heuristic solutions with a good time-quality trade-off.

Overview and Contributions. The remainder is structured as follows. In Section 2 we recapture the original Bus Evacuation Problem (BEP) as given by Bish [4] and show its NP-completeness. We adapt the model to take uncertainty in the number of evacuees into account, and introduce a robust two-stage model, the Robust Bus Evacuation Problem (RBEP). We derive its NP-completeness, and present a mixed-integer linear program (MIP).

We then present solution approaches in Section 3: A linear search over a set of smaller MIPs, and a tabu search approach including the specification of neighborhoods. In Section 4 we proceed to derive analytical lower bounds on the evacuation time, and present computational results comparing our solution approaches with the direct usage of a commercial MIP solver using both randomly generated instances and a real-world instance in Section 5. Finally, we conclude the paper and discuss further research possibilities in Section 6.

2 The Model

2.1 The Nominal Problem

In the following, we will use the terms "collection points" or "sources" for the places where evacuees need to be picked up and "shelters" or "sinks" for the places where they need to be dropped interchangeably. We use the notation $[N]$ for sets $\{1, \ldots, N\}$.

The problem we consider here is to find schedules for a set of buses B such that all evacuees are transported from a set of source nodes S to a set of sink nodes T, minimizing the evacuation time, i.e., the time needed until the last person arrives at a sink. The problem was originally proposed in [4] with slight modifications; in particular, we assume that a bus picks up exactly the number of people that equals its capacity when visiting a source. This assumption was not used in [4], but allows us to consider the demand l_i of a source $i \in [S]$ and the capacity u_j of a sink $j \in [T]$ to be an integer multiple of bus capacities.

At the beginning of the evacuation process, all buses stand at a depot that has a distance of d_i^{start} to the source $i \in [S]$; for further trips we assume a symmetric distance matrix $(d_{ij})_{i\in[S],j\in[T]}$ to be given. Formally, we formulate the Bus Evacuation Problem in the following way:

The Bus Evacuation Problem (BEP):

Input: The number of buses B, of sources S, and of sinks T. A matrix $(d_{ij})_{i\in[S],j\in[T]}$ of source-sink-distances, a vector $(d_i^{start})_{i\in[S]}$ of depot-source-distances, a vector $(l_i)_{i\in[S]}$ of numbers of evacuees, and a vector $(u_j)_{j\in[T]}$ of sink capacities.

Find: Find a tour plan minimizing the maximum travel time over all buses such that all evacuees are transported to the sinks.

We now consider a MIP model for the BEP, that is similar to the one presented in [4]. Table 1 summarizes the variables we use. We choose the concept of *rounds* to model subsequent bus trips; i.e., we estimate the maximum number R of trips a bus needs to do in advance. A trivial way to do so is to set $R = \sum_{i\in[S]} l_i$.

x_{ij}^{br}	Decides if bus b travels from source i to sink j in round r.
t_{to}^{br}	Travel time of bus b in round r from the source to the sink.
t_{back}^{br}	Travel time of bus b in round r from the sink to the next source.

Table 1: Variables of the BEP MIP formulation.

$$\min T \tag{1}$$

$$\text{s.t. } T \geq \sum_{r\in[R]} \left(t_{to}^{br} + t_{back}^{br}\right) + \sum_{i\in[S]}\sum_{j\in[T]} d_i^{start} x_{ij}^{b1} \qquad \forall b \in [B] \tag{2}$$

$$t_{to}^{br} = \sum_{i\in[S]}\sum_{j\in[T]} d_{ij} x_{ij}^{br} \qquad \forall b \in [B], r \in [R] \tag{3}$$

$$t_{back}^{br} \geq d_{ij} \left(\sum_{k\in[S]} x_{kj}^{br} + \sum_{l\in[T]} x_{il}^{b,r+1} - 1 \right) \qquad \forall b \in [B], r \in [R], i \in [S], j \in [T] \tag{4}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br} \leq 1 \qquad \forall b \in [B], r \in [R] \tag{5}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br} \geq \sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{b,r+1} \qquad \forall b \in [B], r \in [R-1] \tag{6}$$

$$\sum_{j\in[T]}\sum_{b\in[B]}\sum_{r\in[R]} x_{ij}^{br} \geq l_i \qquad \forall i \in [S] \tag{7}$$

$$\sum_{i\in[S]}\sum_{b\in[B]}\sum_{r\in[R]} x_{ij}^{br} \le u_j \qquad \forall j \in [T] \tag{8}$$

$$x_{ij}^{br} \in \mathbb{B} \qquad \forall i \in [S], j \in [T], b \in [B], r \in [R] \tag{9}$$

$$t_{to}^{br}, t_{back}^{br} \in \mathbb{R} \qquad \forall b \in [B], r \in [R] \tag{10}$$

$$T \in \mathbb{R} \tag{11}$$

Constraint (2) models the evacuation time as the maximum over the travel times of all buses. Constraints (3) and (4) are used to determine the auxiliary variables t_{to}^{br} and t_{back}^{br}. We ensure that a bus can make only one trip per round with Constraint (5), while Constraint (6) allows a bus to finish early. Finally, Constraints (7) and (8) determine that all persons are evacuated to shelters of sufficient capacity. In this formulation, we assume symmetric distances between the sources and the sinks; note however, that non-symmetric distances could be easily included.

We now discuss the problem complexity.

Theorem 2.1. *BEP is NP-complete, even if $d_i^{start} = 0$ and $d_{ij} = d_{i'j}$ for all $i, i' \in [S]$ and $j \in [T]$.*

Proof. We reduce the problem of scheduling n jobs on P parallel machines ($P||C_{\max}$) to BEP. Due to Garey and Johnson [5], $P||C_{\max}$ is NP-hard.

In a general instance of $P||C_{\max}$, we have P identical machines and n jobs with processing times p_j for all $j = 1, \ldots, P$. Each job can be accomplished by exactly one machine and the processing of a job on a machine can not be interrupted. The problem is to find a assignment of the n jobs to the P machines in order that the maximal completion time $C_{\max}$ is minimized.

For a given $P||C_{\max}$ instance, we construct the corresponding BEP problem in the following way:

- P buses $B = \{B_1, \ldots, B_P\}$
- one source $S = \{s\}$ with $l_s = n + P$
- $n + P$ sinks $T = \{J_1, \ldots, J_n\} \cup \{L_1, \ldots, L_P\}$
  with $u_j = 1 \, \forall j \in T$
  and $d_{sj} = \frac{p_j}{2} \, \forall j \in \{J_1, \ldots, J_n\}$ and $d_{sl} = M \, \forall l \in \{L_1, \ldots, L_P\}$
  where $M = \sum_{j=1}^{P} p_j + 1$

We illustrate this transformation in Figure 1.

For a given instance of $P||C_{\max}$, the corresponding BEP instance can be constructed in polynomial time. We start with some observations on the constructed BEP:

- We have $n + P$ sinks and $n + P$ evacuees in the single source s. Therefore, all sinks must be reached by exactly one bus.
- If there exist a bus $b \in [B]$ which drives to more than than one sink of the set $\{L_1, \ldots, L_P\}$, then $T \geq 3M$.
- If each bus $b \in [B]$ drives to exactly one sink of the set $\{L_1, \ldots, L_P\}$, then $T \leq 2M + \sum_{j=1}^{P} p_j < 3M$. Since we have P sinks of the type $\{L_1, \ldots, L_P\}$ and P buses, it follows that in an optimal plan, each of the P buses drives to exactly one sink of the set $\{L_1, \ldots, L_P\}$.

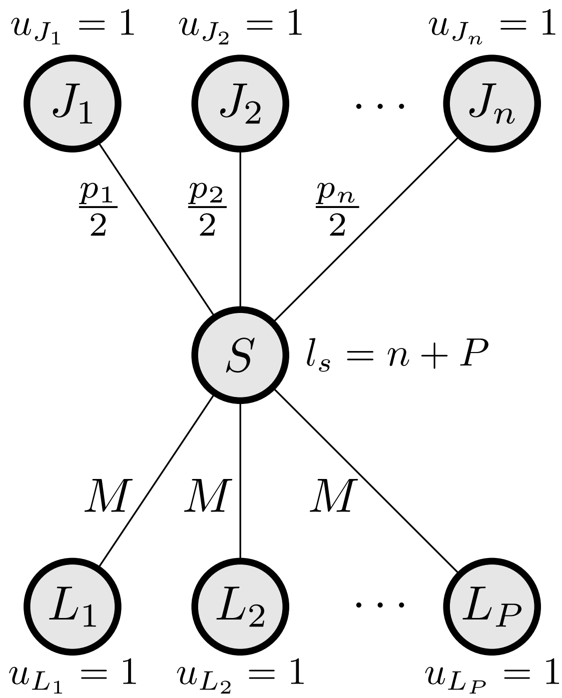

Figure 1: Bus model of $P||C_{\max}$

- If the last and only drive of a bus $b \in [B]$ goes to a sink of the set $\{L_1, \ldots, L_P\}$, then $M \leq T \leq M + \sum_{j=1}^{P} p_j < 2M$.
- If the last drive goes to a sink of the set $\{J_1, \ldots, J_n\}$, then $2M \leq T$.

It follows from these observations, that in any optimal BEP solution, the last drive of each bus goes to a sink of the set $\{L_1, \ldots, L_P\}$. Therefore, we can formulate the following equivalence relation:

"Is there a schedule of $P||C_{\max}$ with $C_{\max} \leq k$?" has a Yes-answer

$$\Leftrightarrow$$

"Is there a driving plan of BEP with $T \leq k + M$?" has a Yes-answer

We have an instance of $P||C_{\max}$ and want to find out if there exists a solution with $C_{\max} \leq k$ for a given k. Let the bus model be constructed in the way we described it before. If we have found a bus plan with $T \leq k + M$, we have a drive plan for each bus $b \in [B]$: $P_b = \{i | J_i \text{ reached by } b\}$ with an additional drive to L_k for exactly one k. The total time for this plan of bus b is $T_b = \sum_{k \in P_b} d_{SJ_k} + M$.

From this plan, we can construct a solution to $P||C_{\max}$ with the optimal value $C_{\max} = k$ in the following way. For each machine $b \in \{1, \ldots, P\}$, process the jobs P_b with the total completion time $\sum_{k \in P_b} p_k$.

This completes the proof that BEP is NP-hard. To show NP-completeness, we note that the completion time

$$T = \max_{B \in [B]} \left\{ \sum_{k \in P_b} d_{sk} \right\}$$

can be checked in polynomial time. □

2.2 The Robust Problem

2.2.1 Problem Specification

The problem we now consider is the following: We assume that the number of evacuees is not known exactly, but we are given a set of estimates for the number of evacuees at each source, i.e., we consider a discrete set of scenarios

$$\mathcal{U} = \{l^1, \dots, l^Z\},$$

where l^i is a vector of length S. After waiting p_{wait} time units, we get the information which of these scenarios was actually realized. For each bus at our disposal, we have to make the following decision:

- Dispatch the bus right now, based on the estimations $\mathcal{U}$. This gives the advantage that the bus does not need to wait the p_{wait} time units; however, there is no exact information on the number of evacuees available. We will refer to these buses as *here-and-now* buses.
- Dispatch the bus after p_{wait} time units, when exact information is available. We will refer to these buses as *wait-and-see* buses.

Once a bus is dispatched, we assume that we cannot change its given tour plan anymore. This assumption is realistic if communication with the buses is not possible, or the evacuation schedule is published and evacuees depend on its adherence.

Formally, the problem we consider is the following:

The Robust Bus Evacuation Problem (RBEP):

Input: The number of buses B, of sources S, of sinks T, and of scenarios Z. A matrix $(d_{ij})_{i\in[S],j\in[T]}$ of source-sink-distances, a vector $(d_i^{start})_{i\in[S]}$ of depot-source-distances, a matrix $(l_i^z)_{z\in[Z],i\in[S]}$ of scenario-dependent numbers of evacuees, a vector $(u_j)_{j\in[T]}$ of sink capacities, and a waiting penalty p_{wait}.

Find: For each bus, decide if it should be detached immediately, or if it should wait. Find a tour plan minimizing the maximum travel time over all buses such that all evacuees are transported to the sinks.

In the following, we illustrate the problem using a small example.

Example 2.1. *We consider the problem instance given in Figure 2. There are three source nodes and two scenarios: In the first scenario, the number of evacuees at the source nodes is* $(1, 2, 1)$*, while in the second scenario it is* $(4, 1, 2)$*. There are three sinks with capacities* $(5, 2, 2)$*. The waiting penalty is* $p_{wait} = 9$*, and there are two buses available.*

We present a solution to the problem in Figure 3. One bus is used as here-and-now, and travels along the route (2–3)–(3–1)–(1–2)–(1–3), i.e., it picks up two bus loads at s_1*, one at* s_2*, and one at* s_3*. The other bus is used as wait-and-see. In the first scenario, it takes the route (3–3)–(2–1), while it takes the route (3–2)–(1–2)–(1–1) in the second scenario.*

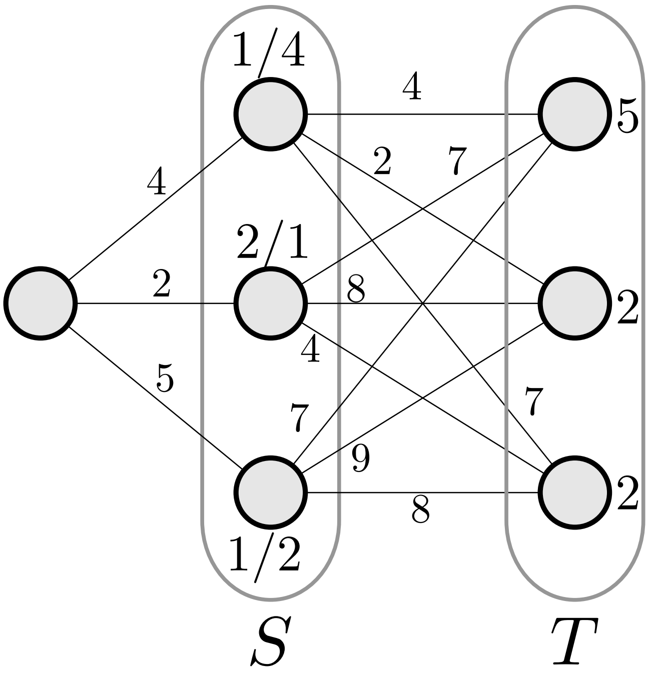

Figure 2: Example instance.

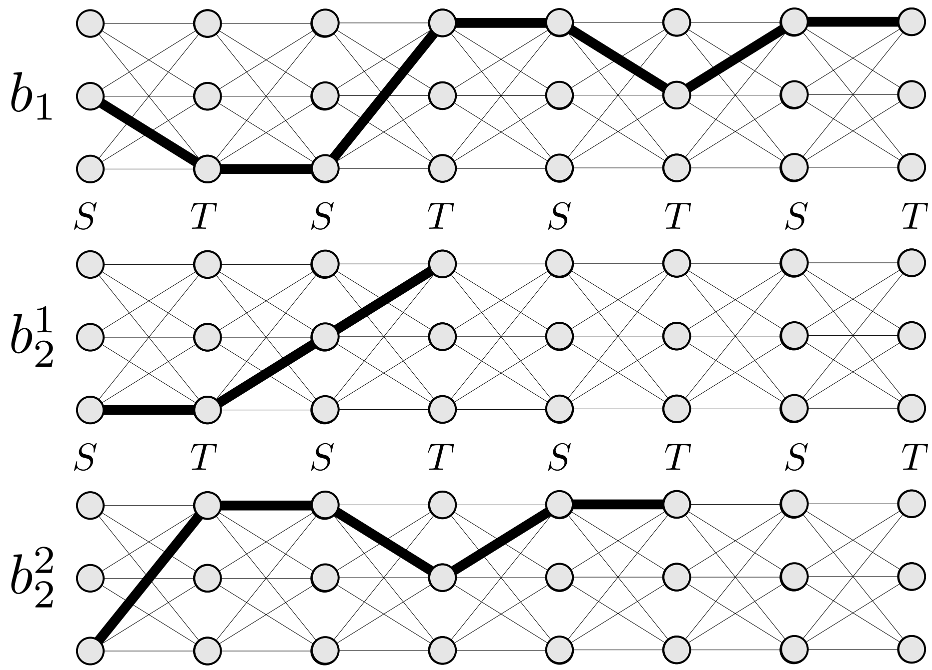

Figure 3: Optimal solution.

The first bus needs a driving time of $2+4+8+7+4+2+2+4=33$, *while the second bus needs time* $9+5+8+4+7=33$ *in the first scenario and time* $9+5+7+4+2+2+4=33$ *in the second scenario. Note that the tour (3–3) was not even necessary for the second bus in the first scenario, as the evacuation demand would have already been fulfilled without it. However, the objective value does not change when this tour is left out; in fact, the presented solution is even optimal.*

Concerning the problem complexity, we can directly use Theorem 2.1 to show the following result.

Theorem 2.2. *RBEP is NP-complete.*

Proof. As BEP is a special case of RBEP with $Z=1$, this follows directly from Theorem 2.1. □

2.2.2 A MIP Formulation

We modify the MIP presented in Section 2.1 to account for the data uncertainty and the possibility to decide whether a bus is here-and-now or wait-and-see. A short description of the variables used in this model is given in Table 2.

$$\min T \tag{12}$$

$$\text{s.t. } T \geq \sum_{r\in[R]} \left(t_{to}^{br} + t_{back}^{br}\right) + \sum_{i\in[S]}\sum_{j\in[T]} d_i^{start} x_{ij}^{b1} \qquad \forall b \in [B] \tag{13}$$

$$T \geq p_{wait}(1-y_b) + \sum_{r\in[R]} \left(t_{to}^{brz} + t_{back}^{brz}\right) + \sum_{i\in[S]}\sum_{j\in[T]} d_i^{start} x_{ij}^{b1z} \qquad \forall b \in [B], z \in [Z] \tag{14}$$

$$t_{to}^{br} = \sum_{i\in[S]}\sum_{j\in[T]} d_{ij} x_{ij}^{br} \qquad \forall b \in [B], r \in [R] \tag{15}$$

$$t_{to}^{brz} = \sum_{i\in[S]}\sum_{j\in[T]} d_{ij} x_{ij}^{brz} \qquad \forall b \in [B], r \in [R], z \in [Z] \tag{16}$$

$$t_{back}^{br} \geq d_{ij}\left(\sum_{k\in[S]} x_{kj}^{br} + \sum_{l\in[T]} x_{il}^{b,r+1} - 1\right) \qquad \forall b \in [B], r \in [R], i \in [S], j \in [T] \tag{17}$$

$$t_{back}^{brz} \geq d_{ij}\left(\sum_{k\in[S]} x_{kj}^{brz} + \sum_{l\in[T]} x_{il}^{b,r+1,z} - 1\right) \qquad \forall b \in [B], r \in [R], i \in [S], j \in [T], z \in [Z] \tag{18}$$

$$\sum_{r\in[R]}\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br} \leq R y_b \qquad \forall b \in [B] \tag{19}$$

$$\sum_{r\in[R]}\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{brz} \leq R(1-y_b) \qquad \forall b \in [B], z \in [Z] \tag{20}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br} \leq 1 \qquad \forall b \in [B], r \in [R] \tag{21}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{brz} \leq 1 \qquad \forall b\in[B], r\in[R], z\in[Z] \tag{22}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{br} \geq \sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{b,r+1} \qquad \forall b\in[B], r\in[R-1] \tag{23}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{brz} \geq \sum_{i\in[S]}\sum_{j\in[T]} x_{ij}^{b,r+1,z} \qquad \forall b\in[B], r\in[R-1], z\in[Z] \tag{24}$$

$$\sum_{j\in[T]}\sum_{r\in[R]}\sum_{b\in[B]} \left(x_{ij}^{br} + x_{ij}^{brz}\right) \geq l_i^z + \sum_{j\in[T]} \Delta_{ij}^z \qquad \forall i\in[S], z\in[Z] \tag{25}$$

$$\sum_{i\in[S]}\sum_{r\in[R]}\sum_{b\in[B]} \left(x_{ij}^{br} + x_{ij}^{brz}\right) \leq u_j + \sum_{s\in[S]} \Delta_{ij}^z \qquad \forall j\in[T], z\in[Z] \tag{26}$$

$$x_{ij}^{br} \in \mathbb{B} \qquad \forall i\in[S], j\in[T], b\in[B], r\in[R] \tag{27}$$

$$x_{ij}^{brz} \in \mathbb{B} \qquad \forall i\in[S], j\in[T], b\in[B], r\in[R], z\in[Z] \tag{28}$$

$$\Delta_{ij}^z \in \mathbb{Z}_+ \qquad \forall i\in[S], j\in[T], z\in[Z] \tag{29}$$

$$y_b \in \mathbb{B} \qquad \forall b\in[B] \tag{30}$$

$$t_{to}^{br}, t_{back}^{br} \in \mathbb{R} \qquad \forall b\in[B], r\in[R] \tag{31}$$

$$t_{to}^{brz}, t_{back}^{brz} \in \mathbb{R} \qquad \forall b\in[B], r\in[R], z\in[Z] \tag{32}$$

x_{ij}^{br}	Decides if the here-and-now bus b travels from source i to sink j in round r.
x_{ij}^{brz}	Decides if the wait-and-see bus b travels from source i to sink j in round r and scenario z.
Δ_{ij}^z	Determines the number of dummy passengers from source i to sink j in scenario z.
y_b	If set to 1, bus b is here-and-now, else it is wait-and-see.
t_{to}^{br}	Travel time of the here-and-now bus b in round r from the source to the sink.
t_{back}^{br}	Travel time of the here-and-now bus b in round r from the sink to the next source.
t_{to}^{brz}, t_{back}^{brz}	Analogously, for the wait-and-see buses.

Table 2: Variables of the RBEP MIP formulation.

Constraints (13) and (14) are used to determine the maximum evacuation time over all here-and-now and wait-and-see buses, respectively. Constraints (15) and (16) model the time needed to travel from sources to sinks, while Constraints (17) and (18) model the time needed back from sinks to sources. Constraints (19) and (20) ensure that a bus is either here-and-now or wait-and-see. It is ensured that every bus can only take one trip per round in Constraints (21) and (22), while Constraints (23) and (24) model that a bus may end its tour, but not restart afterwards. Finally, shelter capacity and that all evacuees are transported in all scenarios are modeled by Constraints (25) and (26)

In the MIP Formulation, we have to include dummy persons. The necessity of this modifications can be seen in the following example.

Example 2.2. *We consider the problem instance given in Figure 4. There are two source nodes s_1, s_2, one sink node t and two scenarios z_1, z_2: In the first scenario, there is one bus load of persons waiting in s_1, and none in s_2. In the second scenario it is the opposite, one bus load of persons is waiting in s_2, and none in s_1. We suppose that both sources can be reached one time unit, i.e. $d_1^{start} = d_2^{start} = 1$. The single sink has a capacity of one, the waiting penalty is $p_{wait} = 1000$ and there is only one bus available.*

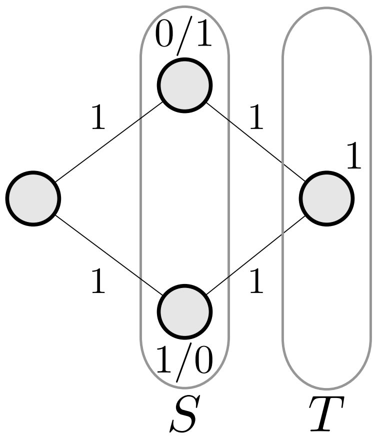

Figure 4: Example instance.

Since p_{wait} is large, it is clear that the bus will not wait and the optimal solution contains the two tours (s_1–t) and (s_2–t) and will pickup the person waiting either in s_1 or s_2. The problem is that there are two trips which end in t, but there is an upper capacity of one such that we have to add dummy passengers.

In the first scenario, the bus does not evacuate a person on the tour (s_2–t) and therefore the dummy passenger is used to modify the upper constraint in u_t. The same can be done for the second scenario.

3 Solution Approaches

3.1 Linear Search

The MIPs presented in the last section have a strong symmetry; in particular, we can permute the set of variables associated with one bus with those of another bus. We can make use of this by fixing the number of buses that are here-and-now and wait-and-see in advance, thus solving $B + 1$ smaller MIPs. For $k \in \{0, \dots, B\}$, we set $C := B \setminus \{1, \dots, k\}$ and $B := \{1, \dots, k\}$, and need solve the following reduced MIP:

$$\min T$$

$$\text{s.t. } T \geq \sum_{r \in [R]} \left(t_{to}^{br} + t_{back}^{br} \right) + \sum_{i \in [S]} \sum_{j \in [T]} d_i^{start} x_{ij}^{b1} \qquad \forall b \in [B]$$

$$T \geq p_{wait} \sum_{r \in [R]} (t^{crz}_{to} + t^{crz}_{back}) + \sum_{i \in [S]} \sum_{j \in [T]} d^{start}_i x^{c1z}_{ij} \qquad \forall c \in [C], z \in [Z]$$

$$t^{br}_{to} = \sum_{i \in [S]} \sum_{j \in [T]} d_{ij} x^{br}_{ij} \qquad \forall b \in [B], r \in [R]$$

$$t^{crz}_{to} = \sum_{i \in [S]} \sum_{j \in [T]} d_{ij} x^{crz}_{ij} \qquad \forall c \in [C], r \in [R], z \in [Z]$$

$$t^{br}_{back} \geq d_{ij} \left(\sum_{k \in [S]} x^{br}_{kj} + \sum_{l \in [T]} x^{b,r+1}_{il} - 1 \right) \qquad \forall b \in [B], r \in [R], i \in [S], j \in [T]$$

$$t^{crz}_{back} \geq d_{ij} \left(\sum_{k \in [S]} x^{crz}_{kj} + \sum_{l \in [T]} x^{crz}_{il} - 1 \right) \qquad \forall c \in [C], r \in [R], i \in [S], j \in [T], z \in [Z]$$

$$\sum_{i \in [S]} \sum_{j \in [T]} x^{br}_{ij} \leq 1 \qquad \forall b \in [B], r \in [R]$$

$$\sum_{i \in [S]} \sum_{j \in [T]} x^{crz}_{ij} \leq 1 \qquad \forall c \in [C], r \in [R], z \in [Z]$$

$$\sum_{i \in [S]} \sum_{j \in [T]} x^{br}_{ij} \geq \sum_{i \in [S]} \sum_{j \in [T]} x^{b,r+1}_{ij} \qquad \forall b \in [B], r \in [R-1]$$

$$\sum_{i \in [S]} \sum_{j \in [T]} x^{crz}_{ij} \geq \sum_{i \in [S]} \sum_{j \in [T]} x^{c,r+1,z}_{ij} \qquad \forall c \in [C], r \in [R-1], z \in [Z]$$

$$\sum_{j \in [T]} \sum_{r \in [R]} \left(\sum_{b \in [B]} x^{br}_{ij} + \sum_{c \in [C]} x^{crz}_{ij} \right) \geq l^z_i + \sum_{j \in [T]} \Delta^z_{ij} \qquad \forall i \in [S], z \in [Z]$$

$$\sum_{i \in [S]} \sum_{r \in [R]} \left(\sum_{b \in [B]} x^{br}_{ij} + \sum_{b \in [C]} x^{crz}_{ij} \right) \leq u_j + \sum_{i \in [S]} \Delta^z_{ij} \qquad \forall j \in [T], z \in [Z]$$

$$x^{br}_{ij} \in \mathbb{B} \qquad \forall i \in [S], j \in [T], b \in [B], r \in [R]$$

$$x^{crz}_{ij} \in \mathbb{B} \qquad \forall i \in [S], j \in [T], c \in [C], r \in [R], z \in [Z]$$

$$\Delta^z_{ij} \in \mathbb{Z}_+ \qquad \forall i \in [S], j \in [T], z \in [Z]$$

$$t^{br}_{to}, t^{br}_{back} \in \mathbb{R} \qquad \forall b \in [B], r \in [R]$$

$$t^{crz}_{to}, t^{crz}_{back} \in \mathbb{R} \qquad \forall c \in [C], r \in [R], z \in [Z]$$

3.2 Tabu Search

The idea of a tabu search heuristic is to keep track of recently visited solutions in a local search, and to avoid visiting the same solution again in order to leave local minima with respect to the neighborhood. In the following we discuss in detail how to apply such a meta-heuristic to the RBEP.

Solution Representation We encode a solution as a table containing lists of tours $T \in [S] \times [T]$. We will refer to a list of tours as a *tourplan* in the following. As an example, the solution from Figure 3 is represented by Table 3.

Note that we do not use the variables for dummy passengers Δ^z_{ij}, which would be possible to implement in the tabu search, but is left out to shrink the

Trip nr.	1	2	3	4
Bus 1	$(2,3)$	$(3,1)$	$(1,2)$	$(1,1)$
Bus 2, Sc 1	$(3,3)$	$(2,1)$		
Bus 2, Sc 2	$(3,1)$	$(1,2)$	$(1,1)$	

Table 3: Representation of the solution given in Figure 3.

search space. To determine the feasibility of a given solution consisting only of tourplans it is possible to solve a matching problem in a bipartite graph for every scenario, in which we assign passengers to bus trips; for speed-up reasons we *do not* solve this problem but choose the conservative evaluation without dummy passengers. This means that the optimal solution to a pathological case like Example 2.2 would be deemed infeasible by our heuristic. We therefore gain a speed-up at the cost of the size of the space of possible solutions.

Neighborhoods We consider the following neighborhoods:

Move 1: Modify Tour. For any tour $T = (s, t)$, change source and target to (s', t'), $s' \in [S], t' \in [T]$.
Move 2: Append tour. Extend the tourplan of any bus by a tour $T = (s, t)$.
Move 3: Delete tour. Remove any tour from the tourplan of any bus. This might actually remove a bus from the current solution.
Move 4: Add bus. Add either a here-and-now bus with a tourplan consisting the tour $T = (s, t)$, or add a wait-and-see bus with a tourplan consisting the tour $T = (s, t)$ for every scenario.
Move 5: Remove bus. Remove a bus including its tourplan from the current solution.
Move 6: Move tour. Move the last tour T of the tourplan of any bus to the end of another tourplan of another bus. This might also create a new here-and-now or wait-and-see bus as in Move 4.
Move 7: Here-and-now to wait-and-see. A here-and-now bus is changed to a wait-and-see bus, and its tourplan is copied for every scenario.
Move 8: Swap tours. Choose any two tours from any two buses, and swap their positions.

Let the objective value of a feasible solution with at least one here-and-now bus be T. Then Move 7 results in a feasible solution with objective value at most $T + p_{wait}$. However, all other moves might make the current feasible solution infeasible; in fact, also searching infeasible regions can be advantageous to escape local minima. We now propose a way to do so.

Further Specifications Also, we propose the followings features:
Infeasible Regions. We allow the search to transfer infeasible regions by a dynamically updated penalty parameter. Specifically, let inf^{sat}, inf^{cap} and inf^{bus} denote the total number of non-evacuated persons over all scenarios, the total number of shelter capacity violations over all scenarios, and the number of buses exceeding B, respectively. To determine the objective value of a solution during the tabu search, we calculate

$$obj_{tabu} := T + p_{sat}\text{inf}^{sat} + p_{cap}\text{inf}^{cap} + p_{bus}\text{inf}^{bus},$$

where p_{sat}, p_{cap} and p_{bus} are penalty parameters. For any sequence of feasible solution, these penalty factors are decreased by a constant factor, while for any sequence of infeasible solutions, these penalty factors are increased.
Tabu List and Resets. The tabu list consists of complete solutions. In every step, we count the number of idle iterations, i.e., the number of iterations where the current best solution was not improved. After a given number *max_idle* of idle iterations, we restart the search, i.e., the tabu list is emptied, infeasibility penalties are reset to their beginning values, and the current best solution is restored.
Domination of New Best Solution. A feasible solution from the neighborhood that improves the current best solution is always preferred over any infeasible solution, however the objective function obj_{tabu} evaluates.
Tour Balancing. From those solutions in the neighborhood that have an equal best objective value, we chose a solution that has the smallest tour length variance, i.e., we apply a lexicographic optimization scheme. Preferring solutions with small tour length variance will results in solutions where buses have tours of balanced lengths.
Randomization. Finally, we choose from those neighboring solutions that evaluate equally well one solution at random.

Constructing Starting Solutions Finally, we need to provide a starting solution for the local search. We follow a simple greedy heuristic that makes uses of two buses: one here-and-now bus that assumes the minimum number of evacuees over all scenarios at each location, and one wait-and-see bus that transports the residual evacuees, depending on the realized scenario. The procedure is presented as Algorithm 1.

In Steps 8–19 we determine a tour for the here-and-now bus b_1 that transports from each source node the minimum number of evacuees over all scenarios, denoted as α_i, $i \in [S]$. In Steps 20–31 the same procedure is repeated for every scenario for the wait-and-see bus b_2, fulfilling the residual demand β_i^z, $z \in [Z]$, $i \in [S]$. Algorithm 1 has polynomial time and space complexity in the input data.

4 Lower Bounds

We present two ways to calculate a lower bound on the objective value of any feasible solution for a given instance of RBEP. Both of them can be calculated in polynomial time, and while the second bound is slightly more elaborate to calculate, it will never be smaller than the first bound.

4.1 LB1

We assume the following simplifications:

1. $d_i^{start} = 0$ for all $i \in [S]$.
2. $p_{wait} = 0$. This allows us to assume w.l.o.g. that all buses are wait-and-see.

Algorithm 1 (Construction of Starting Solution)

Require: An instance of RBEP.

```
b_1 ← ∅
for z ∈ [Z] do
    b_2^z ← ∅
end for
for i ∈ [S] do
    α_i ← min_{z∈[Z]} l_i^z.
end for
for z ∈ [Z] do
    for i ∈ [S] do
        β_i^z ← l_i^z − α_i
    end for
end for
for i ∈ [S] do
    for k = 1, ..., α_i do
        Choose any j ∈ [T] with u_j > 0
        u_j ← u_j − 1
        b_1.pushback((i, j))
    end for
end for
for z ∈ [Z] do
    for j ∈ [T] do
        u_j^z ← u_j
    end for
    for i ∈ [S] do
        for k = 1, ..., β_i^z do
            Choose any j ∈ [T] with u_j^z > 0
            u_j^z ← u_j^z − 1
            b_2^z.pushback((i, j))
        end for
    end for
end for
return Here-and-now tourplan b_1 and vector of wait-and-see tourplans (b_2^z)_{z∈[Z]}.
```

We only try to estimate t^{br}_{to}, i.e., the driving time from the sources to the sinks, ignoring t^{br}_{back}, the driving time for the way back to the sinks. As all buses may be assumed to be wait-and-see, we can consider every scenario separately. For a scenario z and a source node i, we calculate a lower bound on the total time that is needed to evacuate this node by

$$load^z_i := l^z_i \cdot \min_{j\in[T]}\{d_{ij}\}$$

Then, a lower bound on the total amount of driving time in scenario z is given by

$$load^z := \sum_{i\in[S]} load^z_i,$$

and a lower bound on the actual evacuation time by

$$lb^z_1 := \left\lceil \frac{load^z}{B} \right\rceil.$$

The process to calculate the lower bound in summarized in Algorithm 2.

Algorithm 2 (LB1)

Require: An instance of RBEP.
1: **for** $i \in [S]$ **do**
2: $\quad \tau_i \leftarrow \min_{j\in[T]} d_{ij}$.
3: **end for**
4: **for** $z \in [Z]$ **do**
5: $\quad lb^z \leftarrow \left\lceil \sum_{i\in[S]} \frac{l^z_i \cdot \tau_i}{B} \right\rceil$.
6: **end for**
7: $lb_1 \leftarrow \max_{z\in[Z]} lb^z_i$.
8: **return** lb_1

Example 4.1. *We calculate lb_1 for the instance given in Example 2.1. For the first scenario, we find a lower bound $lb^1_1 = \left\lceil \frac{2+8+7}{2} \right\rceil = 9$, and for the second scenario we find $lb^2_1 = \left\lceil \frac{8+4+14}{2} \right\rceil = 13$, yielding $lb_1 = 13$. Recall that the optimal solution has an objective value of 33.*

4.2 LB2

We now extend the ideas that led to the first lower bound in two aspects:

1. When calculating $load^z_i := l^z_i \cdot \min_{j\in[T]}\{d_{ij}\}$ for the previous lower bound, we ignore that sinks have capacities, and the closes sink might not be able to accommodate all evacuees from source i. We can thus improve this estimate by sending only up to u_j units to the closest sink j, then up to $u_{j'}$ units to the second-closest sink j', and so on.

2. In order to estimate t^{br}_{back}, we note that the total number of times a bus visits a source node $i \in [S]$ is at least l^z_i. Of these visits, at most B can come from the depot, while at least $(\sum_{i\in[S]} l^z_i) - B$ visits come from sink nodes $j \in [T]$. Using the minimum distance to any sink for each source, we find a lower bound on the total distance travelled by buses from sink nodes back to source nodes.

We summarize these modification in Algorithm 3. In Steps 4–16 we calculate the lower bound on the total distance travelled by return trips to the sources. We collect all minimum distances according to their multiplicity in the list *backlist*, and remove the B largest distances, assuming that these could be substituted by trips from the depot to the sources. The remaining distances are added to the lower bound.

Algorithm 3 (LB2)

Require: An instance of RBEP.

1: **for** $z \in [Z]$ **do**
2:     $backlist \leftarrow \emptyset$
3:     $lb^z \leftarrow 0$
4:     **for** $s \in [S]$ **do**
5:         $mindist \leftarrow \min_{j\in[T]} d_{ij}$
6:         **for** $k \in 1, \dots, l_i^z$ **do**
7:             $backlist$.pushback($mindist$)
8:         **end for**
9:     **end for**
10:     sort $backlist$
11:     **for** $b \in [B]$ **do**
12:         $backlist$.popback()
13:     **end for**
14:     **for** $v \in backlist$ **do**
15:         $lb^z \leftarrow lb^z + v$
16:     **end for**
17:     **for** $s \in [S]$ **do**
18:         $tolist \leftarrow \emptyset$
19:         **for** $j \in [T]$ **do**
20:             **for** $k = 1, \dots, u_j$ **do**
21:                 $tolist$.pushback(d_{ij})
22:             **end for**
23:         **end for**
24:         sort $tolist$
25:         **for** $k = 1, \dots, l_i^z$ **do**
26:             $lb^z \leftarrow lb^z + tolist$.front()
27:             $tolist$.popfront()
28:         **end for**
29:     **end for**
30: **end for**
31: $lb_2 \leftarrow \left\lceil \frac{\max_{z\in[Z]} lb^z}{B} \right\rceil$
32: **return** lb_2

In Steps 17–29, we calculate a sharpened version of LB1 that takes sink capacities into account. For every source $i \in [S]$, we sort the shortest distances to the sinks and add the shortest ones to the lower bound according to the available capacity, until the number of evacuees at i is met.

Example 4.2. *We compare LB2 to the lower bound from Section 4.1. Estimating the total travel time from source nodes to sink nodes, we calculate for scenario 1 an amount of* $2 + 8 + 7 = 17$*, and for scenario 2 an amount of*

$12 + 4 + 14 = 30$. *For the travel time from sink nodes to source nodes, we find for scenario 1 an amount of* $17 - 7 - 4 = 6$ *and for scenario 2 we get* $26 - 7 - 7 = 12$. *In total, we find a lower bound of*

$$lb_2 = \left\lceil \frac{\max\{17 + 6, 30 + 12\}}{2} \right\rceil = 21$$

Recall that $lb_1 = 13$ *the optimal solution has an objective value of 33.*

5 Experimental Results

5.1 Environment

All experiments were conducted on a compute server with a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 20MB cache, 32 GB RAM and Ubuntu 12.04. We used CPLEX v. 12.4. with OPLRUN for solving MIPs, and gcc v. 4.5.4. with compile flag -O3 for the tabu search.

5.2 Datasets

We ran our experiments on two separate sets of instances:

Randomly Generated Instances For given parameters S, T, B, and Z, draw the parameters at random as shown in Table 4.

variable	d_{ij}	d_i^{start}	l_i^z	u_j	p_{wait}
region	1–10	1–10	1–5	1–10	1–10

Table 4: Instance generation parameters.

This way, we generated 10 instances for each of the parameter sets presented in Table 5, totalling to 70 instances. Infeasible instances were sorted out until the required number of instances was met.

set name	S	T	B	Z
$\mathcal{I}_2$	2	2	2	2
$\mathcal{I}_3$	3	3	2	2
$\mathcal{I}_4$	4	4	3	3
$\mathcal{I}_5$	5	5	3	3
$\mathcal{I}_6$	6	6	4	4
$\mathcal{I}_7$	7	7	4	4
$\mathcal{I}_8$	8	8	5	5

Table 5: Instance sizes.

The City of Kaiserslautern, Germany. We consider the following hypothetical scenario: A military aircraft heading for a large airbase crashes in the city center. It carries a common "general purpose" bomb of type Mark 82 with

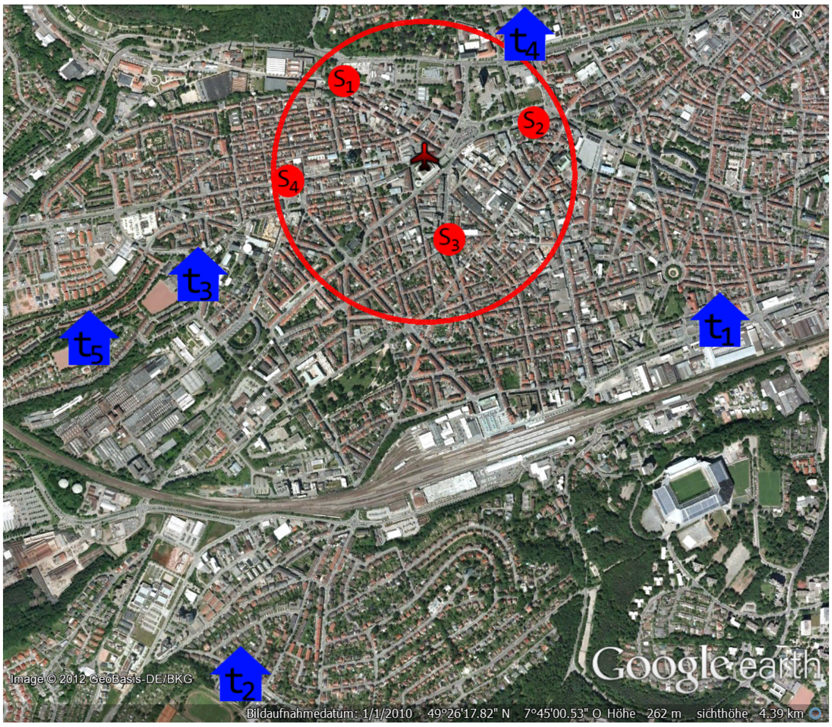

Figure 5: Map of Kaiserslautern (image copyright 2012 Google and 2012 GeoBasis)

a nominal weight of $500lb$, which needs to be defused. According to practitioners, one meter of evacuation radius per pound of explosive is the regular rule of thumb for evacuations, resulting in an evacuation radius of $500m$.

The official statistical record of Kaiserslautern states a population density of about 9000 persons per square kilometer, meaning that about $0.5^2 \cdot \pi \cdot 9000 \approx 7000$ people live within the evacuation region. Since some of the people will go to their family or to friends, we plan to evacuate about 25% of the population to shelters using buses, which are approximately 1750 people. A typical bus as used by the local transport company has a capacity of 80 persons.

To facilitate the evacuee collection, it is quite common to use easily recognizable landmarks as source locations. We assume that evacuees are supposed to meet at four of the largest bus shelters within the evacuation radius.

According to general practice, we assume that evacuees are brought to gymnasiums that get provided with beds, blankets, food, etc. The usual rule-of-thumb is to assume that $3m^2$ are necessary per bed. Table 6 shows the official area of five gymnasiums close to the emergency region with the resulting capacities. A distance matrix is presented in Table 7, where we used the depot of the local transport company as the starting point for the buses.

Shelter	Area in m^2	Capacity in persons	Capacity in bus loads
t_1	400	130	2
t_2	2400	800	11
t_3	1200	400	6
t_4	400	130	2
t_5	800	270	4

Table 6: Shelter capacities.

	t_1	t_2	t_3	t_4	t_5	depot
s_1	6	9	5	4	5	7
s_2	3	7	5	2	6	6
s_3	4	6	4	4	6	7
s_4	7	7	2	6	3	8

Table 7: Distance matrix.

Finally, the considered scenarios are presented in Table 8. In the first scenario we assume an approximately equal distribution of evacuees to sources, and consider four more scenarios, were there is bias towards one source respectively.

	z_1	z_2	z_3	z_4	z_5
s_1	5	9	4	4	3
s_2	6	5	9	5	7
s_3	6	4	6	8	4
s_4	5	3	4	4	8

Table 8: Scenario matrix.

5.3 Setting

For each instance, we ran the following algorithms:

- Solve the MIP formulation with CPLEX using all 16 cores; the timelimit is 180 seconds. We set the CPLEX *mipemphasis* parameter such that the solver focus on improving the current incumbent.
- Use the linear search approach with CPLEX using all 16 cores; the timelimit for each MIP is $180/(B+1)$ seconds. Again, the *mipemphasis* parameter is set to focus on improving the incumbent.
- Run the tabu search with a timelimit of 180 seconds on the 16 cores separately. The respective parameters can be found in Table 9, where p_{incr} and p_{decr} denote the factors penalties are multiplied with during series of feasible/non-feasible solutions, and $\mathcal{I}_n$ denotes the size of the tabu list for instances of category $\mathcal{I}_n$. Over the 16 runs, we choose the best solution per instance.

p_{sat}	p_{cap}	p_{bus}	p_{incr}	p_{decr}			
1	1	2	1.05	0.95			
$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$	$\mathcal{I}_8$	KL
15,000	10,000	5,000	1,000	500	200	100	1,000

Table 9: Tabu search parameters.

The timelimit of 180 seconds was chosen to represent the limited time horizon that is available for planning in an emergency situation. Additionally, we ran CPLEX with a timelimit of 3600 seconds per instance to find better lower bounds for instances that were not solved to optimality (the *mipemphasis* parameter is set to focus on improving the lower bound, accordingly), and calculated LB1 and LB2 for each instance.

5.4 Results

Randomly Generated Instances We present our results in Tables 10 and 11. Every value represents the normalized objective with respect to the best solution per row; i.e., a value of 1.00 corresponds to the best solution, and a lower bound of 1.00 is tight.

For the smallest instances $\mathcal{I}_2$ and $\mathcal{I}_3$, we find that all solution approaches perform about equally well, with a slight edge to CPLEX LS. The lower bound stemming from CPLEX is best, and for 18 out of 20 instances optimality was shown.

The results increasingly diversify for larger instances $\mathcal{I}_4$ to $\mathcal{I}_7$. Using the MIP directly becomes less competitive, and fails to produce solutions from instance size $\mathcal{I}_6$ on. The linear search clearly dominates all other approaches in this instance size region. The lower bound found by Algorithm 3 becomes larger than the one found by CPLEX.

Finally, for the largest instances $\mathcal{I}_8$, also the CPLEX linear search fails to produce feasible solutions, leaving only the tabu search approach as a suitable

	CPLEX MIP	CPLEX LS	Tabu	CPLEX LB	LB1	LB2
$\mathcal{I}_2$-01	1.00	1.00	1.00	1.00	0.50	0.81
$\mathcal{I}_2$-02	1.00	1.00	1.00	1.00	0.46	0.71
$\mathcal{I}_2$-03	1.00	1.00	1.00	1.00	0.25	0.58
$\mathcal{I}_2$-04	1.00	1.00	1.00	1.00	0.40	0.69
$\mathcal{I}_2$-05	1.00	1.00	1.00	1.00	0.25	0.31
$\mathcal{I}_2$-06	1.00	1.00	1.03	1.00	0.39	0.64
$\mathcal{I}_2$-07	1.00	1.00	1.00	1.00	0.56	0.90
$\mathcal{I}_2$-08	1.00	1.00	1.00	1.00	0.50	0.77
$\mathcal{I}_2$-09	1.00	1.00	1.00	1.00	0.39	0.63
$\mathcal{I}_2$-10	1.00	1.00	1.00	1.00	0.44	0.67
Median	**1.00**	**1.00**	**1.00**	**1.00**	**0.42**	**0.68**
$\mathcal{I}_3$-01	1.00	1.00	1.00	0.83	0.51	0.87
$\mathcal{I}_3$-02	1.00	1.00	1.00	1.00	0.40	0.67
$\mathcal{I}_3$-03	1.00	1.00	1.00	1.00	0.39	0.68
$\mathcal{I}_3$-04	1.00	1.00	1.00	1.00	0.37	0.68
$\mathcal{I}_3$-05	1.00	1.00	1.00	1.00	0.48	0.89
$\mathcal{I}_3$-06	1.04	1.00	1.00	0.82	0.45	0.82
$\mathcal{I}_3$-07	1.00	1.00	1.15	1.00	0.35	0.65
$\mathcal{I}_3$-08	1.00	1.00	1.08	1.00	0.55	0.94
$\mathcal{I}_3$-09	1.00	1.00	1.00	1.00	0.48	0.78
$\mathcal{I}_3$-10	1.00	1.00	1.03	1.00	0.49	0.78
Median	**1.00**	**1.00**	**1.00**	**1.00**	**0.46**	**0.78**
$\mathcal{I}_4$-01	1.15	1.00	1.15	0.82	0.42	0.79
$\mathcal{I}_4$-02	1.10	1.00	1.24	0.86	0.33	0.62
$\mathcal{I}_4$-03	1.09	1.00	1.02	0.62	0.32	0.64
$\mathcal{I}_4$-04	1.06	1.00	1.06	0.79	0.35	0.65
$\mathcal{I}_4$-05	1.06	1.00	1.06	0.85	0.44	0.76
$\mathcal{I}_4$-06	1.09	1.00	1.17	0.63	0.34	0.63
$\mathcal{I}_4$-07	1.03	1.00	1.00	0.65	0.48	0.78
$\mathcal{I}_4$-08	1.03	1.03	1.00	0.73	0.43	0.73
$\mathcal{I}_4$-09	1.05	1.05	1.00	0.71	0.41	0.80
$\mathcal{I}_4$-10	1.16	1.00	1.08	0.80	0.40	0.72
Median	**1.07**	**1.00**	**1.06**	**0.76**	**0.41**	**0.73**
$\mathcal{I}_5$-01	1.00	1.00	1.09	0.59	0.41	0.73
$\mathcal{I}_5$-02	1.16	1.05	1.00	0.61	0.39	0.66
$\mathcal{I}_5$-03	1.09	1.00	1.18	0.58	0.42	0.76
$\mathcal{I}_5$-04	1.26	1.00	1.19	0.57	0.26	0.57
$\mathcal{I}_5$-05	1.04	1.00	1.17	0.70	0.39	0.65
$\mathcal{I}_5$-06	1.04	1.00	1.08	0.64	0.32	0.52
$\mathcal{I}_5$-07	1.03	1.00	1.03	0.60	0.46	0.74
$\mathcal{I}_5$-08	1.10	1.00	1.07	0.60	0.40	0.70
$\mathcal{I}_5$-09	1.22	1.00	1.09	0.53	0.44	0.78
$\mathcal{I}_5$-10	1.26	1.06	1.00	0.60	0.40	0.69
Median	**1.10**	**1.00**	**1.09**	**0.60**	**0.40**	**0.69**

Table 10: Results for randomized instances, part one.

	CPLEX MIP	CPLEX LS	Tabu	CPLEX LB	LB1	LB2
$\mathcal{I}_6$-01	-	1.00	1.09	0.27	0.33	0.48
$\mathcal{I}_6$-02	-	1.00	1.36	0.48	0.30	0.52
$\mathcal{I}_6$-03	1.00	1.05	1.73	0.45	0.36	0.59
$\mathcal{I}_6$-04	-	1.00	1.15	0.41	0.31	0.51
$\mathcal{I}_6$-05	1.45	1.00	1.13	0.35	0.39	0.61
$\mathcal{I}_6$-06	-	1.00	1.32	0.45	0.26	0.53
$\mathcal{I}_6$-07	-	1.00	1.06	0.34	0.37	0.63
$\mathcal{I}_6$-08	-	1.00	1.17	0.56	0.25	0.50
$\mathcal{I}_6$-09	-	1.00	1.08	0.13	0.35	0.58
$\mathcal{I}_6$-10	-	1.00	1.64	0.27	0.32	0.59
Median	**-**	**1.00**	**1.16**	**0.38**	**0.33**	**0.55**
$\mathcal{I}_7$-01	-	1.00	1.22	0.29	0.24	0.44
$\mathcal{I}_7$-02	-	1.00	1.02	0.25	0.35	0.59
$\mathcal{I}_7$-03	-	1.00	1.24	0.27	0.30	0.54
$\mathcal{I}_7$-04	-	1.00	1.43	0.25	0.27	0.50
$\mathcal{I}_7$-05	-	1.00	1.41	0.27	0.32	0.59
$\mathcal{I}_7$-06	-	1.00	1.10	0.27	0.24	0.44
$\mathcal{I}_7$-07	-	1.10	1.00	0.48	0.39	0.61
$\mathcal{I}_7$-08	-	1.00	1.25	0.19	0.25	0.48
$\mathcal{I}_7$-09	-	1.00	1.02	0.29	0.20	0.38
$\mathcal{I}_7$-10	-	1.00	1.32	0.32	0.29	0.55
Median	**-**	**1.00**	**1.23**	**0.27**	**0.28**	**0.52**
$\mathcal{I}_8$-01	-	-	1.00	0.11	0.20	0.33
$\mathcal{I}_8$-02	-	1.00	1.58	0.16	0.24	0.37
$\mathcal{I}_8$-03	-	-	1.00	0.22	0.34	0.60
$\mathcal{I}_8$-04	-	-	1.00	0.12	0.21	0.35
$\mathcal{I}_8$-05	-	-	1.00	0.10	0.16	0.27
$\mathcal{I}_8$-06	-	1.00	1.13	0.13	0.24	0.39
$\mathcal{I}_8$-07	-	1.00	1.34	0.14	0.20	0.37
$\mathcal{I}_8$-08	-	1.20	1.00	0.10	0.14	0.22
$\mathcal{I}_8$-09	-	-	1.00	0.13	0.22	0.36
$\mathcal{I}_8$-10	-	1.16	1.00	0.12	0.19	0.34
Median	**-**	**-**	**1.00**	**0.13**	**0.21**	**0.35**

Table 11: Results for randomized instances, part two.

approach for the largest instances due to its good scalability. On these instances, even the simple lower bound 1 produces better results than CPLEX – which is even more significant taken into account that CPLEX was allowed to use all available cores for one hour, while the lower bounds presented in this paper are calculated within milliseconds.

Kaiserslautern Table 12 shows the calculated evacuation times for the Kaiserslautern dataset. We find a similar pattern as for the larger randomized instances, with CPLEX MIP lagging behind both CPLEX LS and the tabu search. The lower bound of 47 minutes found by Algorithm 3 was still better than the one found by CPLEX in one hour of computation time.

CPLEX MIP	CPLEX LS	Tabu	CPLEX LB	LB1	LB2
92	82	81	45	23	47

Table 12: Results for Kaiserslautern.

Table 13 describes the best solution found (with an evacuation time of 81 minutes) in more detail. There are two here-and-now buses, and one wait-and-see bus. Note how most of the tours to the large shelter t_2 that is relatively far away are scheduled for the here-and-now bus, which allows the wait-and-see bus to make short trips to the cloe, smaller shelters to transport the residual evacuees. We suggest that this might make sense for a rule-of-thumb in situations where an operations research approach is not feasible: Use the first trips of the evacuation for shelters that take longer to reach, in order to be able to react faster in upcoming situations.

Trip nr.	1	2	3	4	5	6	7	8
Bus 1	$(2,2)$	$(3,2)$	$(3,5)$	$(4,5)$	$(4,3)$	$(1,1)$	$(2,2)$	$(3,2)$
Bus 2	$(2,2)$	$(3,2)$	$(2,5)$	$(1,5)$	$(4,3)$	$(1,2)$	$(2,2)$	
Bus 3, Sc 1	$(4,3)$	$(1,3)$	$(2,1)$	$(3,3)$	$(3,4)$	$(4,4)$	$(1,3)$	
Bus 3, Sc 2	$(1,1)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,4)$	$(1,2)$		
Bus 3, Sc 3	$(2,4)$	$(2,3)$	$(1,3)$	$(3,4)$	$(2,3)$	$(3,1)$	$(2,3)$	$(4,2)$
Bus 3, Sc 4	$(3,4)$	$(4,4)$	$(3,3)$	$(3,3)$	$(3,3)$	$(3,3)$	$(1,1)$	
Bus 3, Sc 5	$(2,2)$	$(4,4)$	$(4,3)$	$(2,3)$	$(4,3)$	$(4,2)$	$(4,3)$	

Table 13: Best solution found for Kaiserslautern.

6 Conclusion and Further Research

In this work we discussed the problem of evacuating a region with the help of buses. We started from a model that assumes exact knowledge of the number of evacuees at each location, and showed its NP-completeness. We then extended this model to take data uncertainty into account, and modeled the planning possibility to wait for exact data. We discussed solution approaches and presented a tabu search heuristic and lower bounds. In computational experiments on both randomly generated and real-world data we were able to show that it is possible to significantly improve upon the results gained by directly applying a MIP solver.

Many new questions arise. The speed and quality of the proposed lower bounds suggest that a branch and bound approach might be able to further improve upon the presented algorithms. Furthermore, our experimental data shows that the function $f^* : B \to \mathbb{R}$ which maps a number k to the optimal objective value of the resulting reduced problem with fixed number of here-and-now buses k is quasi-convex in nearly all cases (in out of 10,000 tested instances, only 3 of these functions were not quasi-convex). If it was possible to determine quasi-convexity in advance, the linear search proposed in Section 3.1 can be improved to a binary search, thus only $\mathcal{O}(\log B)$ MIPs needed to be solved instead of $\mathcal{O}(B)$. Finally, even more realistic models than the proposed dichotomy between here-and-now and wait-and-see are open research challenges. Additionally, further tests with unsymmetric distances should be done to check if the computations stay similarly complex.

References

[1] Ben-Tal, A., Chung, B.D., Mandala, S.R., Yao, T.: Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. Transportation Research Part B: Methodological **45**(8), 1177 – 1189 (2011)

[2] Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A.: Adjustable robust solutions of uncertain linear programs. Math. Programming A **99**, 351–376 (2003)

[3] Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Mathematics of Operations Research **23**(4), 769–805 (1998)

[4] Bish, D.R.: Planning for a bus-based evacuation. OR Spectrum **33**, 629–654 (2011)

[5] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). W. H. Freeman & Co Ltd (1979)

[6] Goerigk, M., Schöbel, A.: A scenario-based approach for robust linear optimization. In: Proceedings of the First international ICST conference on Theory and practice of algorithms in (computer) systems, TAPAS'11, pp. 139–150. Springer-Verlag, Berlin, Heidelberg (2011)

[7] Hamacher H. W. Tjandra, S.A.: Mathematical modelling of evacuation problems: a state of the art. In: Pedestrian and Evacuation Dynamics, pp. 227–266. Springer, Berlin (2001)

[8] Liebchen, C., Lübbecke, M., Möhring, R.H., Stiller, S.: The concept of recoverable robustness, linear programming recovery, and railway applications. In: R.K. Ahuja, R. Möhring, C. Zaroliagis (eds.) Robust and online large-scale optimization, *Lecture Note on Computer Science*, vol. 5868, pp. 1–27. Springer (2009)

[9] Sayyady, F., Eksioglu, S.D.: Optimizing the use of public transit system during no-notice evacuation of urban areas. Computers & Industrial Engineering **59**(4), 488 – 495 (2010)

[10] Soyster, A.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research **21**, 1154–1157 (1973)

[11] Tuydes H. Ziliaskopoulos, A.: Tabu-based heuristic approach for optimization of network evacuation contraflow. Transportation Research Record: Journal of the Transportation Research Board **1964**, 157–168 (2006)

[12] Xie, C., Turnquist, M.A.: Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. Transportation Research Part C **19**(1), 40 – 63 (2011)

[13] Yao, T., Mandala, S., Chung, B.: Evacuation transportation planning under uncertainty: A robust optimization approach. Networks and Spatial Economics **9**(2), 171–189 (2009)