A Two-Stage Robustness Approach to Evacuation Planning with Buses *

Marc Goerigk[1], Kaouthar Deghdak[2], and Vincent T'Kindt[2]

[1]Fachbereich Mathematik, Technische Universität Kaiserslautern, Germany
[2]Laboratoire d'Informatique, Ecole Polytechnique de l'Université de Tours, France

October 9, 2013

Abstract

We consider the problem of scheduling a bus fleet to evacuate persons from an endangered region. As most of the planning data is subject to uncertainty, we develop a two-stage bicriteria robust formulation, which considers both the evacuation time, and the vulnerability of the schedule to changing evacuation circumstances.

As the resulting integer program is too large to solve it directly using an off-the-shelf solver, we develop an iterative algorithm that successively adds new scenarios to the currently considered subproblem. In computational experiments, we show that this approach is fast enough to deal with an instance modeling an evacuation case within the city of Kaiserslautern, Germany.

1 Introduction

Evacuation planning is an emerging field for the application of operations research methods. For surveys on the field, we refer to [Ham01] and [AG06]. What many such disasters have in common is their no-notice nature, in which exact data is likely to be not available. Thus, evacuation planning under uncertainty is a recently blossoming field of research, see, e.g., [SE10, SHZ09, ZLZW10, YMC09].

Robust optimization (see, e.g., [BTGN09, ABV09]) is one methodology to handle uncertain optimization problems, whose conservative point of view

*Partially supported within the project DSS_Evac_Logistic, by the Federal Ministry of Education and Research Germany as FKZ 13N12229, and by the French National Research Agency as ANR-11-SECU-002-01 (CSOSG 2011).

is very well suited for emergency settings. The recent concepts of adjustable and recoverable robustness (see [LLMS09, BTGGN03]) follow a two-stage approach and allow that a solution solution needs to be only partially fixed before the scenario is revealed. Applications include, e.g., recoverable robust shortest paths [Bü12] and train timetables [LLMS09].

In this work we consider the problem of recoverable robust bus evacuation schedules. Using a simplified version of the model introduced by [Bis11] (see also [GGH13a]), we present a bicriteria model in which a bus schedule is determined that is "easily" updated to account for changes in the input data. To do so, we define a set of possible scenario realizations (the uncertainty set), and a metric that measures the difference of bus schedules.

To solve the resulting large-scale program, we apply a relaxation procedure in which worst-case scenarios are generated iteratively. A similar method has been successfully applied to the problem of min-max regret spanning trees, see [PGAMCVT13, ABV09].

In [GG12], a robust model for the bus evacuation problem with uncertain numbers of evacuees has already been considered. There, a finite scenario set was used, and the scenario is revealed after the first bus trip has been made. For every bus, it needs to be determined, if it waits for the detailed information to be radioed in, or to start right away. Here, we consider a more realistic set of scenarios, and follow the approach that bus schedules can be changed, but these changes should be kept small.

Contributions and overview. The remainder of this work is structured as follows. In Section 2 we present the simplified bus evacuation model in which uncertainty is not yet included. This model is later used as the foundation for the robust model. In Section 3 we introduce the uncertainty set we consider, and the recoverable robust approach that we use. Our robust two-stage bicriteria model is developed in Section 4. Due to the problem complexity, we propose an iterative scenario-generation heuristic to solve the model in Section 5. Further model extensions are discussed in Section 6. In computational experiments modeling the evacuation of Kaiserslautern, Germany, we compare our solutions to a non-robust approach in Section 7. Section 8 concludes this paper and gives pointers for future research.

2 The Simplified Bus Evacuation Problem

We now discuss the basic (non-robust) problem of evacuating an urban region with the help of buses. Throughout the paper we make use of the notation $[N] := \{1, \dots, N\}$. We assume that there is a discrete set of collection points $i \in [S]$ within the endangered region, along with an integer l_i denoting the number of evacuees who need to be picked up. We assume that this number is given in terms of bus loads. There is a discrete set of shelters $j \in [T]$ outside the endangered region, where these evacuees need to be brought. For every shelter, there is a maximum capacity u_j how many busloads can be taken in.

To simplify the model, we further assume that travel time from shelters back to collection points is a constant that is not depending on the collection point; i.e., we estimate these travel times by the time that is needed to get back to a fixed city center. This simplification is reasonable if the travel time between collection points and shelters is much larger than from one collection point to another collection point, that is, when we evacuate a densely populated urban region to shelters that are relatively far away.

Let d_{ij} be the time needed to travel from collection point i to shelter j, and back to the city center. We use a round-based IP formulation as in [GG12], where one "round" corresponds to one trip from collection point to shelter and back to the city center. A trivial bound on the number R of such rounds is $R = \sum_{i\in[S]} l_i$.

We use binary variables x_{ijrb} to determine if bus b goes from collection point i to shelter j in round r, and a variable $C_{\max}$ that models the maximum travel time over all buses. The basic, non-robust problem is then modeled using the following linear integer program:

$$
\begin{aligned}
(P) \quad & \min C_{\max} && && (1)\\
& \sum_{r\in[R]} \sum_{i\in[S]} \sum_{j\in[T]} d_{ij} x_{ijrb} \le C_{\max} && \forall b \in [B] && (2)\\
& \sum_{i\in[S]} \sum_{j\in[T]} x_{ijrb} \le 1 && \forall b \in [B], r \in [R] && (3)\\
& \sum_{j\in[T]} \sum_{r\in[R]} \sum_{b\in[B]} x_{ijrb} \ge l_i && \forall i \in [S] && (4)\\
& \sum_{i\in[S]} \sum_{r\in[R]} \sum_{b\in[B]} x_{ijrb} \le u_j && \forall j \in [T] && (5)\\
& x \in \{0,1\}^{S\times T\times R\times B} && && (6)\\
& C_{\max} \in \mathbb{N} && && (7)
\end{aligned}
$$

Our objective (1) is to minimize the evacuation time, that is, the maximum of all bus travel times. Constraints (2) are used to measure these travel times. Constraints (3) ensure that every bus can only make on trip per round; and Constraints (4) and (5) ensure that every evacuee is picked up, and shelter capacities are respected.

Example 1. *We present a simple example instance for P in Figure 1.*

There are two collection points s_1 and s_2, and two shelters t_1 and t_2. We assume that there are 4 bus loads of evacuees waiting to be picked up at s_1, and 2 loads at s_2; i.e., $l_1 = 4$ and $l_2 = 2$. For the sake of simplicity, we

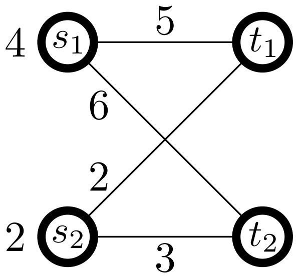


Figure 1: Example instance for P.

ignore shelter capacities in this example. The distances are given as $d_{11} = 5$, $d_{12} = 6$, $d_{21} = 2$, *and* $d_{22} = 3$.

Assuming there are two buses available, the following is an optimal solution to this problem: Both buses perform the operations $s_1 \to t_1$, $s_1 \to t_1$, *and* $s_2 \to t_1$. *The evacuation time* C_{max} *is 12 for this solution.*

3 Data Uncertainty and Robust Optimization

3.1 Definition of Uncertainty

In the following, we consider in which ways uncertain data may affect the basic model P. We assume that our planning takes place during some time when no evacuation is necessary, and all data are only expert estimates. This means that these estimates will not be accurate during an actual emergency; a more realistic approach is therefore to define a *set* $\mathcal{U}$ of possible scenarios that should be considered during the planning process. Specifically, in the above model, uncertainty will affect:

- the travel times d_{ij}, i.e., some trips may take longer due to obstructions or traffic;
- the numbers of evacuees l_i, i.e., evacuees may distribute differently, less or more may wish to use the bus;
- the number of buses B, i.e., some buses may be demolished due to the catastrophe, or being repaired; and
- the shelter capacities u_j, i.e. estimates may be wrong, buildings may be damaged.

Obviously, there are different ways to include such uncertainties. Here we follow the approach that the number of evacuees, the travel times, and the number of buses are uncertain at the same time, and each parameter comes from a given interval, i.e., we have $d_{ij} \in \{\underline{d}_{ij}, \ldots, \overline{d}_{ij}\}$, $l_i \in \{\underline{l}_i, \ldots, \overline{l}_i\}$, and

$B \in \{\underline{B}, \ldots, \overline{B}\}$. Note that the integrality of the values d, l and B lead to a *finite*, but large uncertainty set.

Furthermore, as assuming these values to be independent of each other would lead to unrealistically extreme cases (e.g., $B = \underline{B}$, $d_{ij} = \overline{d}_{ij}$, and $l_i = \overline{l}_i$), we add a constraint to the uncertainty set $\mathcal{U}$ of the form

$$\frac{C^{\text{exp}}_{\max}}{\frac{1}{S \cdot T}\sum_{i,j} d_{ij}} B \geq \sum_i l_i$$

where $\frac{1}{S \cdot T}\sum_{i,j} d_{ij}$ is the average travel time of a trip, and $C^{\text{exp}}_{\max} \in \mathbb{R}$ is an estimated upper bound on the evacuation time. In other words, we only consider instances where the estimated duration of the evacuation process

$$\frac{(\sum_i l_i) \cdot (\sum_{i,j} d_{ij})}{S \cdot T \cdot B}$$

is less or equal to some bound $C^{\text{exp}}_{\max}$. This way, we can ignore extreme cases where an evacuation is likely to take extraordinary long time.

This is in the same spirit as the well-known Γ-robustness approach by [BS03, BS04], where extreme scenarios of an interval-based uncertainty are removed from the scenario set by considering only those scenarios where a up to Γ many coefficients deviate from their expected value at the same time. Adding such a constraint reduces unnecessary over-conservatism of a robust model, as the scenario set becomes more realistic.

Overall, we may write the considered uncertainty set $\mathcal{U}$ as

$$\mathcal{U} = \Big\{(l, d, B) \ : l_i \in \{\underline{l}_i, \ldots, \overline{l}_i\}, d_{ij} \in \{\underline{d}_{ij}, \ldots, \overline{d}_{ij}\}, B \in \{\underline{B}, \ldots, \overline{B}\}, \\ \frac{C^{\text{exp}}_{\max}}{\frac{1}{S \cdot T}\sum_{i,j} d_{ij}} B \geq \sum_i l_i \Big\}. \tag{8}$$

As usual, we will write $P(\mathcal{U})$ to denote the uncertain optimization problem with uncertainty set $\mathcal{U}$. Furthermore, we assume the existence of a *nominal* scenario $(\hat{l}, \hat{d}, \hat{B}) \in \mathcal{U}$ that denotes the most likely scenario (i.e., the scenario one might consider in P if there was no uncertainty set). We shall write $\xi = (l, d, B)$ to denote a scenario; and $\hat{\xi}$ to denote the nominal scenario. Furthermore, we will refer to d^ξ as the vector of distances in scenario ξ, and to l^ξ and B^ξ analogously.

3.2 Robust Counterparts

Given an uncertain optimization problem $P(\mathcal{U})$, there are many different ways to formulate a so-called robust counterpart, that is, a deterministic optimization problem that takes the uncertainty set $\mathcal{U}$ into account. Robust optimization, contrary to stochastic optimization, does not make any assumptions on probability distributions over the input data.

In *strict robustness*, the basic and oldest approach, which goes back to [Soy73] and the seminal work of [BTN98, BTN99, BTN00], one aims at finding a solution that is a) feasible under all possible scenarios, and b) optimizes the worst-case performance.

As such an approach often yields solutions that are too conservative for real-world applications, recent concepts have been developed in which the feasibility restriction is relaxed. One such approach is recoverable robustness (see [LLMS09, Sti08, EMS09]), which is a two-stage concept: We consider a (set of) recovery algorithm(s) that take a solution and modify it to become feasible in a given scenario. We aim at finding a solution for which a) these changes are moderate, and b) the resulting worst-case performance is good.

In the following, we will formalize this approach.

4 A Robust Bicriteria Two-Stage Model

We now propose modifications of the nominal model P that take the uncertainty $\mathcal{U}$ into account. To this end, we introduce sets of second-stage variables x^ξ for every scenario $\xi \in \mathcal{U}$, and demand that our main solution and the scenario-dependant solutions must be "similar". Of course, different definitions how to measure the difference between two solutions are possible. In the following, we focus on the following definition:

$$\Delta : \{0,1\}^{STRB} \times \{0,1\}^{STRB} \to \mathbb{N}$$
$$(x, x^\xi) \mapsto \sum_{i\in[S]} \sum_{j\in[T]} \sum_{r\in[R]} \sum_{b\in[B]} \left| x_{ijrb} - x^\xi_{ijrb} \right|,$$

that is, we count how often a bus makes a different trip in a round.

Finding a first-stage solution x with second-stage solutions x^ξ that minimize the value $\max_{\xi\in\mathcal{U}} \Delta(x, x^\xi)$ amounts to finding bus schedules that are easily recoverable in practice; i.e., if a bus driver finds a different situation during the evacuation process than expected, he only needs to do small modifications in his schedule to be feasible again. For the first-stage solution, we demand feasibility in the nominal scenario $(\hat{l}, \hat{d}, \hat{B})$.

The resulting problem is hence a bicriteria problem; while we would like to minimize the worst-case evacuation time over all scenarios on the one hand, we would also like to minimize the difference between the first- and second-stage solutions. Note that on the on hand, by adding many redundant tours in each scenario, it is possible to get a very small recovery distance at the cost of high evacuation times; on the other hand, taking bus schedules in each scenario minimizing the respective evacuation time will give a small worst-case evacuation time, but high recovery costs. Thus, both objective functions are in conflict.

Furthermore, note that we can refine the criterion of worst-case evacuation time even further: We may consider both the nominal evacuation time

$C_{\max}^{\text{nom}}$, i.e., the duration of the first-stage solution under nominal conditions, and the worst-case evacuation time $C_{\max}^{\text{wc}}$, i.e., the worst possible evacuation time of all second-stage solutions. In practice, however, one would couple both criteria, expressing one criterion as a function of the other one. As an example, one would like the worst-case evacuation time to be at most 50% larger than the nominal evacuation time. Thus, we will consider the evacuation time as a single criterion in the following.

In the field of multicriteria optimization, many methods are known how to determine the set of *Pareto* solutions, which are not dominated in one of the objectives. In this work we follow the ε-constraint method and consider single-criterion problems with an additional constraint ensuring a desired quality on the other objective. From a practical perspective, it is desired to ensure a certain evacuation time quality, and to find bus schedules minimizing the recovery distance under these circumstances. The reason for this is two-fold: Firstly, the evacuation time should have top priority, and a given limit on it should not be crossed. Secondly, while the evacuation time is a very descriptive value, the recovery distance is a more abstract value, and fixing a desired quality is hardly possible in practice.

In the following, we will denote this problem as $RP(\mathcal{U})$.

We now discuss an integer linear programming formulation for $RP(\mathcal{U})$, based on the program for P. As before, we use variables x_{ijrb} denoting if bus b goes from source i to shelter j in round r for the first-stage schedule. For the scenario-dependant second-stage schedules, we use corresponding variables x^{ξ}_{ijrb} for every $\xi \in \mathcal{U}$. There are several further auxiliary variables being used: The variables δ_b and δ^{ξ}_b determine if bus $b \in \overline{B}$ is used in the first- or second-stage schedule, respectively. This is necessary, as different numbers of buses are available in each scenario. Moreover, variables y^{ξ}_{ijrb} are used to determine if the variables x_{ijrb} and x^{ξ}_{ijrb} differ, and thus help determining the value of the variable Δ, which is the recovery distance.

Finally, we need variables z_{ijrb} and z^{ξ}_{ijrb} that determine if a trip x_{ijrb} or x^{ξ}_{ijrb}, respectively, actually transports any evacuees, or is empty. This may happen due to the usage of redundant trips to reduce the number of necessary modifications to make a first-stage solution feasible. However, such redundant trips should not be counted when the feasibility with respect to shelter capacity requirements is concerned.

The full IP we consider is the following:

$$\min \Delta \tag{9}$$

$$\sum_{r\in[R]} \sum_{i\in[S]} \sum_{j\in[T]} \hat{d}_{ij} x_{ijrb} \leq C_{\max}^{\text{nom}} \qquad \forall b\in[\overline{B}] \tag{10}$$

$$\sum_{i\in[S]} \sum_{j\in[T]} x_{ijrb} \leq 1 \qquad \forall b\in[\overline{B}], r\in[R] \tag{11}$$

$$\sum_{j\in[T]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} z_{ijrb} \geq \hat{l}_i \qquad \forall i\in[S] \tag{12}$$

$$\sum_{i\in[S]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} z_{ijrb} \leq u_j \qquad \forall j\in[T] \tag{13}$$

$$\sum_{r\in[R]}\sum_{i\in[S]}\sum_{j\in[T]} d^{\xi}_{ij} x^{\xi}_{ijrb} \leq C^{\text{wc}}_{\max} \qquad \forall b\in[\overline{B}],\xi\in\mathcal{U} \tag{14}$$

$$\sum_{i\in[S]}\sum_{j\in[T]} x^{\xi}_{ijrb} \leq 1 \qquad \forall b\in[\overline{B}],r\in[R],\xi\in\mathcal{U} \tag{15}$$

$$\sum_{j\in[T]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} z^{\xi}_{ijrb} \geq l^{\xi}_i \qquad \forall i\in[S],\xi\in\mathcal{U} \tag{16}$$

$$\sum_{i\in[S]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} z^{\xi}_{ijrb} \leq u_j \qquad \forall j\in[T],\xi\in\mathcal{U} \tag{17}$$

$$x_{ijrb} \leq \delta_b \qquad \forall i\in[S],j\in[T],r\in[R],b\in[\overline{B}] \tag{18}$$

$$x^{\xi}_{ijrb} \leq \delta^{\xi}_b \qquad \forall i\in[S],j\in[T],r\in[R],b\in[\overline{B}],\xi\in\mathcal{U} \tag{19}$$

$$\sum_{b\in[\overline{B}]} \delta_b \leq \hat{B} \tag{20}$$

$$\sum_{b\in[\overline{B}]} \delta^{\xi}_b \leq B^{\xi} \qquad \forall \xi\in\mathcal{U} \tag{21}$$

$$-y^{\xi}_{ijrb} \leq x_{ijrb} - x^{\xi}_{ijrb} \leq y^{\xi}_{ijrb} \qquad \forall i\in[S],j\in[T],r\in[R],b\in[\overline{B}],\xi\in\mathcal{U} \tag{22}$$

$$\Delta \geq \sum_{i\in[S]}\sum_{j\in[T]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} y^{\xi}_{ijrb} \qquad \forall \xi\in\mathcal{U} \tag{23}$$

$$\tag{24}$$

$$z_{ijrb} \leq x_{ijrb} \tag{25}$$

$$z^{\xi}_{ijrb} \leq x^{\xi}_{ijrb} \tag{26}$$

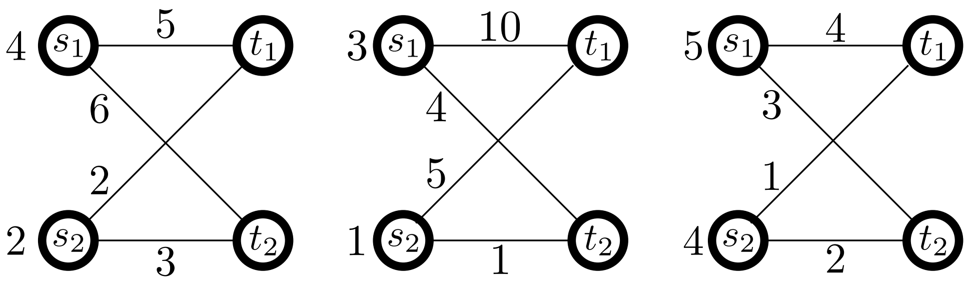
$$\tag{27}$$

$$x, z, x^{\xi}, z^{\xi}, y^{\xi} \in \mathbb{B}^{ST\overline{B}R} \qquad \forall \xi\in\mathcal{U} \tag{28}$$

$$\delta, \delta^{\xi} \in \mathbb{B}^{\overline{B}} \qquad \forall \xi\in\mathcal{U} \tag{29}$$

$$\Delta \in \mathbb{N} \tag{30}$$

We now explain these constraints in more detail. Constraints (10)-(13) encapsule a standard problem of type P, where $C^{\text{nom}}_{\max}$ is not a variable anymore, but a given constant. They ensure that the first-stage solution is feasible; i.e., the evacuation time is sufficiently small, all evacuees are transported, and



(a) Nominal scenario. (b) Scenario 1. (c) Scenario 2.

Figure 2: Example instance for *RP*.

shelter capacities are respected. The following Constraints (14)-(17) do the same for the respective second-stage variables. We ensure that the correct number of buses is used in each scenario with the help of Constraints (18)-(21).

Furthermore, the Constraints (22) and (23) couple the first- and second-stage variables. The former forces the variables y^{ξ}_{ijrb} to be one if and only if the respective first- and second-stage variables differ. The latter ensures that Δ is at least the recovery distance in each scenario; as the objective is to minimize Δ, it will be equal to $\max_{\xi\in\mathcal{U}} \Delta(x, x^{\xi})$ in an optimal solution.

Finally, Constraints (25) and (26) ensure that empty trips are only possible, if a trip is scheduled at all.

Note that, as $\mathcal{U}$ is a finite set, the above IP formulation is finite. However, $|\mathcal{U}|$ is still too large for a commercial IP-solver to be likely to solve the problem directly on instances of real-world size. This motivates the discussion of alternative solution approaches, as will be introduced in the following section.

We conclude this section by revisiting the example of Section 2.

Example 2. *We consider an uncertain version of the nominal instance presented in Example 1. There are two additional scenarios, as presented in Figure 2: In scenario 1, some distances are strongly increased, but the number of evacuees decreased; in scenario 2 on the other hand, distances are decreased, and the number of evacuees increased. In both scenarios, we assume the number of buses to stay* $B = 2$*.*

Consider again the nominal solution as given by Table 1.

	Round 1	Round 2	Round 3
Bus 1	$s_1 \to t_1$	$s_1 \to t_1$	$s_2 \to t_1$
Bus 2	$s_1 \to t_1$	$s_1 \to t_1$	$s_2 \to t_1$

Table 1: Nominal solution.

How does this solution perform under the given uncertainty? Assume *that the worst-case evacuation time C^{wc}_{max} is bounded by 20 (recall that the nominal solution has a nominal evacuation time of 12).*

In the first scenario, all evacuees are still being picked up; thus, an infeasibility can only stem from a violation of the worst-case evacuation time. And, in fact, the evacuation time of the nominal solution in scenario 1 is 25, which is larger than allowed. Thus, the first-stage solution needs to be modified to become a feasible second-stage solution.

The upper part of Table 2 shows one possibility to do so, using $\Delta = 2$ modifications in the schedule. The resulting evacuation time is 20 in this scenario.

		Round 1	Round 2	Round 3	Round 4	Round 5
Scenario 1	Bus 1	$s_1 \to t_1$	–	$s_2 \to t_1$		
	Bus 2	$s_1 \to t_1$	$s_1 \to t_1$	–		
Scenario 2	Bus 1	$s_1 \to t_1$	$s_1 \to t_1$	$s_2 \to t_1$	$s_2 \to t_1$	$s_2 \to t_1$
	Bus 2	$s_1 \to t_1$	$s_1 \to t_1$	$s_2 \to t_1$	$s_1 \to t_2$	

Table 2: Recovered nominal solutions.

Concerning scenario 2, we also note that the nominal first-stage solution becomes infeasible, this time due to an insufficient number of picked up evacuees. The lower part of Table 2 shows a corresponding second-stage solution that is feasible, and needs an evacuation time of 12; as three trips are added, Δ is increased to be 3.

Can we find a solution that needs less modifications to be feasible in each of these scenarios? Assume that we are allowed to increase the evacuation time in the nominal scenario to be at most 16. We present a robust first-stage solution in Table 3.

	Round 1	Round 2	Round 3	Round 4
Bus 1	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$
Bus 2	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$

Table 3: Robust solution.

Note how the choice of routes is more diversified than in the nominal solution, and two redundant trips are added. As before, we evaluate its performance in the two considered Scenarios; recovered solutions are presented in Table 4.

In scenario 1, no modifications are needed, as the evacuation time is 20. In scenario 2, one trip needs to be added to become feasible. Thus, in total, we find that the robust solution has $\Delta = 1$.

		Round 1	Round 2	Round 3	Round 4	Round 5
Scenario 1	Bus 1	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$	
	Bus 2	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$	
Scenario 2	Bus 1	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$	$s_1 \to t_2$
	Bus 2	$s_1 \to t_1$	$s_1 \to t_2$	$s_2 \to t_1$	$s_2 \to t_2$	

Table 4: Recovered robust solutions.

5 An Iterative Solution Approach

Throughout this section, we will simplify the notation for problem RP and denote first- and second-stage soulutions only by x and x^ξ, respectively, neglecting the variables z and z^ξ determining if trips are empty or not. This is only for ease of presentation, and the inclusion of empty trips is straightforward.

5.1 Oracle-Based Algorithms

Throughout this section, we assume that all scenarios $\xi \in \mathcal{U}$ are feasible, i.e., we define the set of feasible solution in scenario ξ as

$$\begin{aligned}
\mathcal{F}(\xi) = \Big\{ x^\xi \in \mathbb{B}^{ST\overline{B}R} : & \sum_{r\in[R]}\sum_{i\in[S]}\sum_{j\in[T]} d^\xi_{ij} x^\xi_{ijrb} \le C^{\text{wc}}_{\max} && \forall b\in[\overline{B}] \\
& \sum_{i\in[S]}\sum_{j\in[T]} x^\xi_{ijrb} \le 1 && \forall b\in[\overline{B}], r\in[R] \\
& \sum_{j\in[T]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} x^\xi_{ijrb} \ge l^\xi_i && \forall i\in[S] \\
& \sum_{i\in[S]}\sum_{r\in[R]}\sum_{b\in[\overline{B}]} x^\xi_{ijrb} \le u_j && \forall j\in[T] \\
& x^\xi_{ijrb} \le \delta^\xi_b && \forall b\in[\overline{B}] \\
& \sum_{b\in[\overline{B}]} \delta^\xi_b \le B^\xi && \Big\}
\end{aligned}$$

and assume them to be non-empty. Note that this is the case, if and only if $RP(\mathcal{U})$ is feasible.

In the following, we present a heuristic procedure that sequentially generates scenarios and adds them to the main problem. To this end, we consider subsets of scenarios $\mathcal{U}' \subseteq \mathcal{U}$. In particular, let $RP(\mathcal{U}')$ denote problem RP using the uncertainty set $\mathcal{U}'$. We will denote a solution that is only defined for the scenarios $\mathcal{U}'$ by $(x, x^\xi)_{\xi\in\mathcal{U}'}$.

We begin by considering $\mathcal{U}' = \emptyset$, i.e., $RP(\emptyset)$ is equivalent to the original bus evacuation problem as described in Section 2, except that the evacuation time is constant. Solving it results in a first solution $(x^{(0)}, x^{(0),\xi})_{\xi \in \mathcal{U}'}$. We then add more scenarios iteratively, with the help of an *oracle*:

Definition 3. *For a given problem $RP(\mathcal{U})$, an* oracle Ω *is a function that takes an uncertainty set $\mathcal{U}' \subseteq \mathcal{U}$ and a solution $(x, x^{\xi})_{\xi \in \mathcal{U}'}$ and returns a scenario $\xi = \Omega((x, x^{\xi})_{\xi \in \mathcal{U}'}) \in \mathcal{U}$ or "No" when this is not possible.*

Assume now there is an oracle available that generates a worst-case scenario $\xi \in \mathcal{U}$ for the current solution $(x^{(k)}, x^{(k),\xi})_{\xi \in \mathcal{U}^{(k)}}$. We add such a scenario to $\mathcal{U}^{(k)}$, and get the new uncertainty set $\mathcal{U}^{(k+1)} = \mathcal{U}^{(k)} \cup \{\xi\}$. Resolving $RP(\mathcal{U}^{(k+1)})$, we get a new solution $(x^{(k+1)}, x^{(k+1),\xi})_{\xi \in \mathcal{U}^{(k+1)}}$. This can be repeated until no more scenarios need to be added. Algorithm 1 summarizes this approach.

```
Algorithm 1
Require: A problem instance $RP(\mathcal{U})$, and an oracle $\Omega$.
Ensure: A solution $(x, x^{\xi})_{\xi \in \mathcal{U}'}$ for a set $\mathcal{U}' \subseteq \mathcal{U}$.
 1: Set $k = 0$, and $\mathcal{U}^{(0)} = \emptyset$.
 2: Solve $RP(\mathcal{U}^{(k)})$. Let $(x^{(k)}, x^{(k),\xi})_{\xi \in \mathcal{U}^{(k)}}$ be the resulting solution, and
    $\Delta^{(k)}$ the resulting recovery distance.
 3: Set $k = k + 1$.
 4: if $\Omega((x^{(k-1)}, x^{(k-1),\xi})_{\xi \in \mathcal{U}^{(k-1)}}) \neq$ "No" then
 5:     Set $\mathcal{U}^{(k)} = \mathcal{U}^{(k-1)} \cup \Omega((x^{(k-1)}, x^{(k-1),\xi})_{\xi \in \mathcal{U}^{(k-1)}})$.
 6:     Solve $RP(\mathcal{U}^{(k)})$. Let $(x^{(k)}, x^{(k),\xi})_{\xi \in \mathcal{U}^{(k)}}$ be the resulting solution, and
    $\Delta^{(k)}$ the resulting recovery distance.
 7:     Goto Step 3
 8: else
 9:     return $(x^{(k-1)}, x^{(k-1),\xi})_{\xi \in \mathcal{U}^{(k-1)}}$
10: end if
```

Note that Algorithm 1 always terminates, as the set of scenarios $\mathcal{U}$ is finite. To show convergence to an optimal solution, we need to refine the concept of an oracle to produce scenarios that are "bad" for the current solution.

Definition 4. *Let a desired recovery distance Δ be given. We call an oracle Ω a* worst-case *oracle if for all $\mathcal{U}' \subseteq \mathcal{U}$ and $(x, x^{\xi})_{\xi \in \mathcal{U}'}$ the following holds for the generated scenario ξ: There is no solution $x^{\xi} \in \mathcal{F}(\xi)$ for scenario ξ with $\Delta(x, x(\xi)) \leq \Delta$. A worst-case oracle returns "No" if such a scenario does not exist.*

We now show that the solution of Algorithm 1 can be extended to an optimal solution of $RP(\mathcal{U})$ without increasing the recovery distance, if the oracle used is a worst-case oracle.

Theorem 5. *If Ω is a worst-case oracle with respect to $\Delta^{(k-1)}$ in Steps 4 and 5, Algorithm 1 produces a solution that can be extended to an optimal solution of RP($\mathcal{U}$) without increasing the recovery distance.*

Proof. Let Ω be a worst-case oracle with respect to $\Delta^{(k-1)}$ in each iteration, and let $(x^{(k^*)}, x^{(k^*),\xi})_{\xi\in\mathcal{U}^{(k^*)}}$ be the solution produced by Algorithm 1. We show that there is an extension $(x^{(k^*)}, x^{(k^*),\xi})_{\xi\in\mathcal{U}}$ of this solution to the whole set of scenarios $\mathcal{U}$, that does not increase its objective value, i.e.,

$$\max_{\xi\in\mathcal{U}^{(k^*)}} \Delta(x^{(k^*)}, x^{(k^*),\xi}) = \max_{\xi\in\mathcal{U}} \Delta(x^{(k^*)}, x^{(k^*),\xi})$$

Assume that there is some $\xi \in \mathcal{U} \setminus \mathcal{U}^{(k^*)}$ with $\min_{x^\xi} \Delta(x, x^\xi) \geq \Delta^{(k^*)}$, where the minimum is taken over all feasible solutions in scenario ξ. Then, as Ω is a worst-case oracle, Algorithm 1 would not have terminated after Step 4 by definition.

On the other hand, $RP(\mathcal{U}')$ is a relaxation of $RP(\mathcal{U})$, and thus the optimal objective value of the former is less or equal to the optimal objective value of the latter. □

The crucial point of Algorithm 1 is to have an effective means for computing a worst-case oracle. In the following, we will consider how this can be done.

5.2 Computing Worst-Case Oracles

To construct a worst-case scenario, we need to find a scenario ξ for a given first-stage solution x for which all feasible solutions have a recovery distance that is larger than the current best. A sufficient condition for such a scenario is be that it is an optimal solution to the following problem, which we will call WC:

$$\max_{\xi\in\mathcal{U}} \min_{x^\xi\in\mathcal{F}(\xi)} \Delta(x, x^\xi)$$

This is a non-linear bi-level problem (where the non-linearity is present in the uncertainty set $\mathcal{U}$). In the following, we will develop an iterative solution method to this subproblem of Algorithm 1. Note that we might try to enumerate all scenarios ξ, but this is a computationally impractical choice. Instead, we will follow the following approach: For a given scenario, we calculate a solution $x(\xi)$ minimizing the recovery distance; and for a given set of precomputed candidate solutions $\{x(\xi), \xi \in \mathcal{U}'\}$ we check if there is a scenario ξ' for which all candidate solutions are infeasible. As we will see, such a method finds an optimal solution to WC.

We will now describe this iterative approach in more detail. First, assume that a scenario $\xi \in \mathcal{U}$ and a first-stage solution x are fixed. We then determine the minimal recovery distance by solving the following problem, which we denote by WC_1:

$$\min \sum_{i\in[S]} \sum_{j\in[T]} \sum_{r\in[R]} \sum_{b\in[\overline{B}]} y^{\xi}_{ijrb} \tag{31}$$

$$\sum_{r\in[R]} \sum_{i\in[S]} \sum_{j\in[T]} d^{\xi}_{ij} x^{\xi}_{ijrb} \leq C^{\text{wc}}_{\max} \qquad \forall b \in [\overline{B}] \tag{32}$$

$$\sum_{i\in[S]} \sum_{j\in[T]} x^{\xi}_{ijrb} \leq 1 \qquad \forall b \in [\overline{B}], r \in [R] \tag{33}$$

$$\sum_{j\in[T]} \sum_{r\in[R]} \sum_{b\in[\overline{B}]} x^{\xi}_{ijrb} \geq l^{\xi}_i \qquad \forall i{\in}[S] \tag{34}$$

$$\sum_{i\in[S]} \sum_{r\in[R]} \sum_{b\in[\overline{B}]} x^{\xi}_{ijrb} \leq u_j \qquad \forall j{\in}[T] \tag{35}$$

$$x^{\xi}_{ijrb} \leq \delta^{\xi}_b \qquad \forall i{\in}[S], j{\in}[T], r{\in}[R], b{\in}[\overline{B}] \tag{36}$$

$$\sum_b \delta^{\xi}_b \leq B^{\xi} \tag{37}$$

$$-y^{\xi}_{ijrb} \leq x_{ijrb} - x^{\xi}_{ijrb} \leq y^{\xi}_{ijrb} \qquad \forall i{\in}[S], j{\in}[T], r{\in}[R], b{\in}[\overline{B}] \tag{38}$$

$$x^{\xi}, y^{\xi} \in \mathbb{B}^{ST\overline{B}R} \tag{39}$$

$$\delta^{\xi} \in \mathbb{B}^{\overline{B}} \tag{40}$$

WC_1 is NP-hard, as it contains the decision problem of P as a subproblem. As before, we use variables x^{ξ}_{ijrb} to model the recovered bus schedule, variables y^{ξ}_{ijrb} to measure the recovery distance, and variables δ^{ξ}_b to count the number of buses. Note that the additional variable Δ is not necessary here, as only a single scenario is considered, and the recovery distance can be directly written in the objective function (31). Constraints (32)-(37) ensure the feasibility of the recovered solution, i.e., that $x^{\xi} \in \mathcal{F}(\xi)$. Finally, Constraints (38) are used to determine the difference between the recovered solution and the original first-stage solution.

Now, assume that a set of candidate solutions $\{x(\xi), \xi \in \mathcal{U}'\}$ is fixed, that all have a recovery distance less or equal to the given value of Δ. If there is no scenario in $\mathcal{U}$ for which such all these solutions are infeasible, we can conclude that there is no solution to the worst-case oracle; i.e., we return ”No”. If, on the other hand, such a scenario ξ exists, we will use it to add a new solution $x(\xi)$ to the set of candidate solutions. If no such solution exists, we can return ξ as a worst-case scenario.

Let a set $x(\xi_k), k = 1, \ldots, N$ of candidate solutions be given. We need to find a scenario $\xi \in \mathcal{U}$ such that all solutions $x(\xi_k)$ are infeasible for this scenario, i.e., $x(\xi_k) \notin \mathcal{F}(\xi)$. We denote this problem by WC_2, and first show that it is NP-hard.

Theorem 6. *WC_2 is NP-hard for an uncertainty set of the form (8), even if $\underline{B} = \overline{B}$ and $\underline{l}_i = \bar{l}_i$ for all $i \in [S]$.*

Proof. We show NP-hardness by using a reduction from Set Cover: Given a set $\mathcal{S} = [s]$ and a finite set of subsets $\mathcal{M}_i \subseteq \mathcal{S}$, $i \in [m]$, decide if there is a choice of subsets $\mathcal{I} \subseteq [m]$ with $\cup_{i \in \mathcal{I}} \mathcal{M}_i = \mathcal{S}$ and $|\mathcal{I}| \leq K$. Without loss of generality, we assume that $\mathcal{M}_i \neq \emptyset$ for all $i \in [m]$.

We show that we can construct an instance of WC_2 that is feasible (i.e., it is possible to find a scenario that is infeasible for all given bus schedules) if and only if Set Cover allows a feasible solution. Example 7 illustrates the following formal description.

To this end, we consider an instance with $S = 1$, $T = m$, and define the following uncertainty set:

$$\mathcal{U} = \left\{ (l, d, B) : l_1 = 1, B = m, d_{ij} \in \{0, 1\}, \frac{C^{\text{exp}}_{\text{max}} \cdot S \cdot T \cdot B}{l_1} \geq \sum_{i,j} d_{ij} \right\},$$

where $C^{\text{exp}}_{\text{max}} = K/m^2$, i.e., the nonlinear constraint in $\mathcal{U}$ becomes $\sum_{ij} d_{ij} \leq K$. We set $C^{\text{wc}}_{\text{max}} = 1/2$, and assume that shelter capacities are sufficiently large (e.g., $u_j = m$ for all $j \in [T]$).

For element $j \in \mathcal{S}$, we generate a bus schedule $x(\xi_j)$ in the following way: A bus $i \in [m]$ makes a single trip from the collection point to shelter $i \in [m]$, if $j \in \mathcal{M}_i$. If, on the other hand, $j \notin \mathcal{M}_i$, then bus i does not make any trip at all.

All these bus schedules are feasible with respect to the number of buses used (which is at most m), and the number of busloads evacuated (as $\mathcal{M}_i \neq \emptyset$ for all $i \in [m]$). The only way to find a scenario $\xi \in \mathcal{U}$ such that these bus schedules are infeasible, is to violate the constraints on the maximum allowed evacuation time (which is 1/2).

If a solution to this instance of WC_2 can be found, we can directly transform it to a solution of Set Cover by setting $i \in \mathcal{I}$ if and only if $d_{1i} = 1$. If this instance of WC_2 is infeasible, then also Set Cover must be infeasible. As all steps require only polynomial time and space, WC_2 is NP-hard.

□

Example 7. *Let an instance of Set Cover be given with the following properties:*

$$\begin{aligned} \mathcal{S} &= \{1, 2, 3, 4\} \\ \mathcal{M}_1 &= \{1, 2\} \\ \mathcal{M}_2 &= \{1, 3\} \\ \mathcal{M}_3 &= \{2, 3, 4\} \\ K &= 2 \end{aligned}$$

We create an instance of WC_2, where we use $B = 3$ buses, $S = 1$ collection point, and $T = 3$ shelters. Also, there is one busload of evacuees waiting at the collection point, and $C^{exp}_{max} = 2/9$, $C^{wc}_{max} = 1/2$. We consider the following four bus schedules:

For x_1, bus 1 goes to shelter 1, and bus 2 goes to shelter 2 (as $1 \in \mathcal{M}_1$, and $1 \in \mathcal{M}_2$). For x_2, bus 1 goes to shelter 1, and bus 3 goes to shelter 3. For x_3, bus 2 goes to shelter 2, and bus 3 goes to shelter 3. Finally, for x_4, bus 3 goes to shelter 3.

To find a scenario that is simultaneously infeasible for the schedules x_1, x_2, x_3, and x_4, we need to find values $d_{1i} \in \{0, 1\}$, $i \in \{1, \dots, 3\}$, such that:

$$
\begin{aligned}
& d_{11} \geq \frac{1}{2} \quad \textit{or} \quad d_{12} \geq \frac{1}{2} \\
& \textit{and} \\
& d_{11} \geq \frac{1}{2} \quad \textit{or} \quad d_{13} \geq \frac{1}{2} \\
& \textit{and} \\
& d_{12} \geq \frac{1}{2} \quad \textit{or} \quad d_{13} \geq \frac{1}{2} \\
& \textit{and} \\
& d_{13} \geq \frac{1}{2}
\end{aligned}
$$

and the constraint $\sum_i d_{1i} \leq 2$ is ensured. A feasible solution is given by setting $d_{11} = 0$, $d_{12} = 1$, and $d_{13} = 1$, which corresponds to picking sets 2 and 3 for the Set Cover problem.

Note that NP-hardness of WC_2 still holds if both $C^{\text{wc}}_{\text{max}}$ and $C^{\text{exp}}_{\text{max}}$ are required to be integer. To this end, only technicalities of the above proof needed to be changed: An additional shelter is added, with travel times to this shelter being always zero. Furthermore, l_1 is not required to be 1, but $m(m+1)$ instead. Every bus will need to make $m+1$ trips to the additional shelter. This way, $C^{\text{exp}}_{\text{max}}$ can be set to be equal to K. Furthermore, to ensure integrality of $C^{\text{wc}}_{\text{max}}$, every bus would be required to make not only one trip if $j \in \mathcal{M}_i$, but two of such trips. Then, setting $C^{\text{wc}}_{\text{max}} = 1$ yields the required proof. For the ease of presentation, these aspects are not included in the proof of Theorem 6.

Problem WC_2 can be solved using the following IP model, where the nonlinearity in $\mathcal{U}$ needs to be carefully reformulated using the integrality of the variables l and d.

$$
\begin{aligned}
\max\ & z && (41) \\
& \underline{l}_i \leq l_i \leq \bar{l}_i && {\scriptstyle \forall i \in [S]} \quad (42) \\
& \underline{d}_{ij} \leq d_{ij} \leq \bar{d}_{ij} && {\scriptstyle \forall i \in [S], j \in [T]} \quad (43)
\end{aligned}
$$

$$\underline{B} \leq B \leq \overline{B} \tag{44}$$

$$z \leq s^{\xi_k} \qquad \forall k \in [N] \tag{45}$$

$$s^{\xi_k} \leq l_i - \sum_{j\in[T]} \sum_{r\in[R]} \sum_{b\in[\overline{B}]} x^{\xi_k}_{ijrb} + M(1 - c^{\xi_k,1}_i) \qquad \forall i \in [S], k \in [N] \tag{46}$$

$$s^{\xi_k} \leq \sum_{i\in[S]} \sum_{j\in[T]} \sum_{r\in[R]} d_{ij} x^{\xi_k}_{ijrb} - C^{\text{wc}}_{\max} + M(1 - c^{\xi_k,2}_b) \qquad \forall b \in \overline{B}, k \in [N] \tag{47}$$

$$s^{\xi_k} \leq B(k) - B + M(1 - c^{\xi_k,3}) \qquad \forall k \in [N] \tag{48}$$

$$\sum_{i\in[S]} c^{\xi_k,1}_i + \sum_{b\in[\overline{B}]} c^{\xi_k,2}_b + c^{\xi_k,3} = 1 \qquad \forall k \in [N] \tag{49}$$

$$\gamma_{ii'jkk'} \geq \alpha_{ik} + \beta_{i'jk'} - 1 \qquad \forall i,i' \in [S], j \in [T], k, k' \tag{50}$$

$$\sum_k 2^k \alpha_{ik} = l_i \qquad \forall i \in [S] \tag{51}$$

$$\sum_k 2^k \beta_{ijk} = d_{ij} \qquad \forall i \in [S], j \in [T] \tag{52}$$

$$c \cdot B \geq \sum_{i,i',j,k,k'} 2^{k+k'} \gamma_{ii'jkk'} \tag{53}$$

$$c^{\xi,1}, c^{\xi,2}, c^{\xi,3} \in \{0,1\} \tag{54}$$

$$\gamma, \alpha, \beta \in \{0,1\} \tag{55}$$

$$s^{\xi}, l, d, B \in \mathbb{N} \tag{56}$$

where

$$B(k) := \left| \left\{ b \in [\overline{B}] : \exists i \in [S], j \in [T], r \in [R] : x^{\xi}_{ijrb} > 0 \right\} \right|$$

denotes the number of buses used in solution x^{ξ}_{k}.

We now explain this formulation in more detail. The variables l, d, and B model the scenario we try to determine. The variables $s(\xi_k)$ model the infeasibility of solution $x(\xi_k)$. It is larger than zero if and only if there is at least one constraint concerning $x(\xi_k)$ that is violated, i.e., $x(\xi_k) \notin \mathcal{F}(l, d, B)$. $c^{\xi_k,1}$, $c^{\xi_k,2}$, and $c^{\xi_k,3}$ are boolean variables that are used to determine which constraint becomes violated. Finally, variables α, β, and γ are used to linearize the nonlinear constraint in the uncertainty set $\mathcal{U}$. α is used for a binary representation of the integer l, β for d, and γ for their product. As these values are bounded, so is also the number of variables α, β, and γ. We can easility calculate this value using the logarithm of base 2.

In other words, for every scenario we need to determine one constraint that is violated, and we want to maximize the minimal violation.

5.3 The Complete Algorithm

We now describe how to combine the above IP models to solve $RP(\mathcal{U})$. We begin with an empty scenario set, and determine the solution of $RP(\emptyset)$. We then proceed to generate a scenario for which all current first- and second-stage solutions are infeasible, using WC_2, and add it to $\mathcal{U}'$. If no such scenario exists, we have found an optimal solution and end the algorithm. If, on the other hand, such a scenario exists, then we check using WC_1 if we can repair the current first-stage solution and keep the same objective value of Δ as before. If this is the case, then we proceed to generate the next scenario (i.e., we the current first-stage solution is kept). If this is not the case, we have to solve $RP(\mathcal{U}')$ again, and generate new first- and second-stage solutions.

This procedure is summarized as Algorithm 2.

Algorithm 2 Iterative Solution Method to $RP(\mathcal{U})$.

```
Require: A problem instance $RP(\mathcal{U})$, and an oracle $\Omega$.
 1: Set $\mathcal{U}' = \emptyset$.
 2: Solve $RP(\mathcal{U}')$.
 3: Solve Problem $WC_2$ using the solution from Step 2.
 4: if the objective value of Step 3 is positive then
 5:     add the generated scenario $\xi$ to $\mathcal{U}'$
 6:     Solve $WC_1$.
 7:     if the objective value is greater than the current $\Delta$ then
 8:         Return to Step 2.
 9:     else
10:         Return to Step 3.
11:     end if
12: else
13:     return optimal solution found.
14: end if
```

As we will discuss in the following section, the time-consuming part of this algorithm is to solve $RP(\mathcal{U}')$, even for relatively small scenario sets. Both WC_1 and WC_2 are solved with considerably less computational effort. So, to improve the performance of Algorithm 2, we will only add scenarios to $\mathcal{U}'$ as described in Step 5, if the current first-stage solution is not feasible for the current value of Δ (i.e., if the condition in Step 7 is fulfilled). This way, the total number of iterations may increase, but the problems in Step 2 are kept as small as possible.

6 Model Extensions

We now discuss further extensions of the robust model RP.

1. **Return travel time.** The first possible extension considers the nominal model we use. In Section 2, a model is presented where the travel time from shelter locations back to collection points only depends on the shelter, not the collection point under consideration. A more realistic approach would consider travel times that also depend on the collection point that a bus travels to. This can be done by considering two sets of x variables: $\overrightarrow{x}_{ijrb}$ to model that bus b goes from collection point $i \in [S]$ to shelter $j \in [T]$ in round $r \in [R]$; and $\overleftarrow{x}_{ijrb}$ to model that bus b goes from shelter $j \in [T]$ to collection point $i \in [S]$ in round $r \in [R]$.

   The adaptation of the constraints in model P to this approach is straightforward. Additionally, we need to ensure that the number of trips from collection points to shelters, and from shelters to collection points keep in equilibrium by using

   $$\sum_{i\in[S]}\sum_{j\in[T]}\sum_{r\in[R']} \overrightarrow{x}_{ijrb} \geq \sum_{i\in[S]}\sum_{j\in[T]}\sum_{r\in[R']} \overleftarrow{x}_{ijrb} \qquad \forall b \in [B], R' \in [R]$$

   $$\sum_{i\in[S]}\sum_{j\in[T]}\sum_{r\in[R']} \overleftarrow{x}_{ijrb} + 1 \geq \sum_{i\in[S]}\sum_{j\in[T]}\sum_{r\in[R']} \overrightarrow{x}_{ijrb} \quad \forall b \in [B], R' \in [R]$$

   The basic ideas of the presented methods are still applicable: For a given set of first- and second-stage solution, we generate scenarios for which these solutions are infeasible, and add those scenarios to the robust formulation, as described in Algorithm 2.

2. **Time-dependant travel times.** In an emergency setting, travel times might change over time. Using a different nominal model that is based on a time-discretisation (instead of a round-discretisation), time-dependant values for d_{ij} can be included. As before, with a corresponding adaptation of the WC_1 and WC_2 models, we will still be able to solve the resulting robust problem using Algorithm 2.

3. **Other uncertainty sets.** We can easily include different models for $\mathcal{U}$ than presented in Equation 8: The only condition necessary for Algorithm 2 is that we are still able to solve WC_2. To his end, obviously any polyhedron can be used, and any nonlinear constraint that can be linearized.

7 Experiments

In the following, we present an experimental evaluation of the model and the oracle-based algorithm. We begin with describing the instance used, then describe experimental setup, and finally discuss the results gained.

Datasets. We consider the following instance, which is modeled after the city of Kaiserslautern, Germany (see also [GG12, GGH13b]). The setting is a $500lb$ bomb defusion within the city center, leading to an evacuation radius of approximately $500m$. The total number of evacuees who make use of the evacuation by bus is about 1750 people; with a bus capacity of 80, this corresponds to a total of 22 bus loads. These are distributed over four collection points within the affected region. Additionally, there are four shelter locations available, which are gymnasiums of different capacities. Three buses are available.

To account for the uncertainty in this instance, we constructed three modified instances of varying degree of uncertainty. For the instance with *small* uncertainty, we assume that every trip may take one minute more or less (i.e., $d_{ij} \in \{\hat{d}_{ij} - 1, \hat{d}_{ij} + 1\}$), and every collection point may have one bus load of evacuees more or less (i.e., $l_i \in \{\hat{l}_i - 1, \hat{l}_i + 1\}$). Furthermore, one bus may fail.

For the second instance, where we consider a *medium* degree of uncertainty, this is modified so that there may be up to two additional bus loads of evacuees at the collection points, i.e., $l_i \in \{\hat{l}_i - 1, \hat{l}_i + 2\}$.

Finally, for the last instance with a *large* degree of uncertainty, we extend this to be $l_i \in \{\hat{l}_i - 2, \hat{l}_i + 2\}$. All other parameters are as before.

Setup. We want to use the proposed iterative Algorithm 2 to solve the above three uncertain evacuation problems. In preliminary experiments, we identified the main computational problem to be the solution of $RP(\mathcal{U}')$ for increasing sizes of scenario sets $\mathcal{U}'$. Both problems WC_1 and WC_2 are relatively easy to solve in comparison.

For this reason, we considered an alternative approach to solve $RP(\mathcal{U}')$ without using an IP solver for the model as-is. In the spirit of [FM13], we rewrite the model as a feasibility problem, considering the objective function as a hard constraint. Beginning with the objective value of the last solution gained in the iterative process, we check if a solution with the same objective value exists. If not, it is increased by one, and the solution process is repeated.

To speed up the solution of these feasibility problems, we use a proximity objective function; i.e., the difference to the last-known solution. This way, the IP solver uses an improved branching tree compared to using simply 0 as an objective function, and prefers solutions that are similar to the last solution. In the following, we refer to the approach of using the IP model of $RP(\mathcal{U}')$ directly as "IP", while referring to the approach using linear feasibility search as "Loc" (for "local search").

Each of the three instances is solved with each of these methods with three different parameters for $C^{\text{nom}}_{\max}$ and $C^{\text{wc}}_{\max}$. These parameters are:

$$C^{\text{nom}}_{\max} = 1.00 \cdot C^*_{\max} \text{ and } C^{\text{wc}}_{\max} = 1.50 \cdot C^*_{\max};$$

$$C_{\max}^{\text{nom}} = 1.25 \cdot C_{\max}^{*} \text{ and } C_{\max}^{\text{wc}} = 1.75 \cdot C_{\max}^{*};$$
$$C_{\max}^{\text{nom}} = 1.50 \cdot C_{\max}^{*} \text{ and } C_{\max}^{\text{wc}} = 2.00 \cdot C_{\max}^{*},$$

where $C_{\max}^{*}$ denotes the optimal nominal objective value.

For each of these solutions, we measured: The objective value Δ, the solution time, the total number of scenarios that were generated, and the size of $\mathcal{U}'$ for the largest subproblem $RP(\mathcal{U}')$ considered. Furthermore, we also considered the robust objective value Δ of the nominal solution. Note that to compute this value, scenarios need to be generated in the manner of Algorithm 2.

Environment. All experiments were conducted on an Intel Xeon CPU E5-2670 running with 2.60GHz. Only one of 16 cores were used for each run. A random access memory of 96GB was available, but all runs needed less than one GB of memory. Programs were compiled under Ubuntu using gcc 4.6.3, and IPs solved with Cplex 12.4.

Results. We present the measured results in Tables 5, 6, and 7. All instances were solved to optimality.

In Table 5 we compare the robust objective values Δ of the robust solution, and the nominal solution. It can be seen that in this case, allowing the same nominal budget $C_{\max}^{\text{nom}}$ for the robust solution as the nominal solution needs leads to no improvement in the robustness (i.e., in the rows with $C_{\max}^{\text{nom}} = 1.00$, the objective values of the robust and the nominal solutions are the same). In other words, there was no way to find "robustness for free" on these instances, that is, to improve the robustness without losing nominal quality.

Furthermore, as expected, the robust objective value decreases with increasing size of $C_{\max}^{\text{nom}}$ and $C_{\max}^{\text{wc}}$. For the small- and medium-sized instances, *strictly* robust solutions (i.e., solutions where no modifications are necessary to be feasible in any scenario) were already found from $C_{\max}^{\text{nom}} = 1.25$ on.

In Table 6 we compare the computation times of Algorithm 2 using IP and using Loc. We find that in all but one cases, the proximity search was able to solve the robust problems faster. Also, consider the significant increase in computation time from the medium-sized instances to the large instances. Naturally, increasing the size of the possible intervals for each value l_i leads to an exponential increase of possible scenarios, which leads to more iterations in Algorithm 2.

Finally, Table 7 gives the number of scenarios generated in the solution process (column "All"), and the largest number of scenarios considered when solving $RP(\mathcal{U}')$ (column "WC"). We see that in correspondence to Table 6, the largest instances have significantly more scenarios that are necessary to consider, even though the number of worst-case scenarios WC is only slightly increased. In other words, even though many more scenarios need

size	$C_{\max}^{\text{nom}}$	$C_{\max}^{\text{wc}}$	Robust	Nominal
small	1.00	1.50	1	1
	1.25	1.75	0	1
	1.50	2.00	0	1
medium	1.00	1.50	1	1
	1.25	1.75	0	1
	1.50	2.00	0	1
large	1.00	1.50	2	2
	1.25	1.75	1	2
	1.50	2.00	0	2

Table 5: Robust objective values Δ.

size	$C_{\max}^{\text{nom}}$	$C_{\max}^{\text{wc}}$	IP	Loc
small	1.00	1.50	1.68	1.14
	1.25	1.75	10.96	1.82
	1.50	2.00	9.19	1.62
medium	1.00	1.50	4.46	2.21
	1.25	1.75	9.15	2.76
	1.50	2.00	8.54	2.80
large	1.00	1.50	778.69	660.11
	1.25	1.75	457.61	480.23
	1.50	2.00	638.29	379.67

Table 6: Computation times.

size	$C_{\max}^{\text{nom}}$	$C_{\max}^{\text{wc}}$	IP WC	IP All	Loc WC	Loc All
small	1.00	1.50	1	4	1	4
	1.25	1.75	4	4	4	4
	1.50	2.00	4	6	4	4
medium	1.00	1.50	1	4	1	4
	1.25	1.75	4	4	4	4
	1.50	2.00	4	4	4	4
large	1.00	1.50	1	70	1	80
	1.25	1.75	5	66	5	69
	1.50	2.00	4	66	5	68

Table 7: Number of scenarios.

to be considered, the size of the robust problem $RP(\mathcal{U}')$ does not significantly increase.

To summarize the presented results, we argue that a) the proposed robust model is able to produce solutions with a reasonable trade-off between nominal and robust objective value, which allows the planner a larger flexibility in his spectrum of possible evacuation plans; b) solution times are still within a desired window for preemptive planning; and c) solution times are very sensitive to the size of the uncertainty set.

8 Conclusion and Further Research

In this work we considered a robust bicriteria model for the regional evacuation with the help of buses, where one criterion is its worst-case or nominal evacuation time, and the other criterion is its robustness, measured as the number of modifications in the bus schedule which are necessary to be feasible in any scenario in the worst-case. Its applicability is increased by its possibility to handle complex, non-linear uncertainty sets.

We presented an iterative solution algorithm that alternately solves the robust problem for a limited set of scenarios, and generates new scenarios from the uncertainty set. Such an algorithm can be easily extended to other uncertain optimization problems, and other uncertainty sets. The result is the whole Pareto front of solutions with respect to the evacuation time and the robustness. We presented computational experience on an instance modeling the evacuation of the city of Kaiserslautern, and showed that the robust problem can be solved within reasonable computation times.

Further lines of research are plenty: On the computational side, is it possible to generate scenarios in WC_2, where we know in advance that the current solution cannot be repaired within the same budget Δ? Such an approach would reduce the total number of generated scenarios, and thus the computation times. Can we use advanced IP techniques as branch-and-price to solve $RP(\mathcal{U}')$ even faster? On the modeling side, how well do the robust solutions behave under simulated, random delays? Can we include reconnaissance aspects, so that the degree of uncertainty can be reduced for a given cost? Can we use preprocessing techniques to reduce the size of the uncertainty? All these questions can be considered in further research.

References

[ABV09] H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427 – 438, 2009.

[AG06] Nezih Altay and Walter G. Green III. OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475 – 493, 2006.

[Bis11] D. R. Bish. Planning for a bus-based evacuation. *OR Spectrum*, 33:629–654, 2011.

[BS03] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:2003, 2003.

[BS04] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[BTGGN03] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Programming A*, 99:351–376, 2003.

[BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, Princeton and Oxford, 2009.

[BTN98] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[BTN99] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.

[BTN00] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming A*, 88:411–424, 2000.

[Bü12] Christina Büsing. Recoverable robust shortest path problems. *Networks*, 59(1):181–189, 2012.

[EMS09] A.L. Erera, J.C. Morales, and M. Savelsbergh. Robust optimization for empty repositioning problems. *Operations Research*, 57(2):468–483, 2009.

[FM13] M. Fischetti and M. Monaci. Proximity search for 0-1 mixed-integer convex programming. Technical report, Università degli Studi di Padova, 2013.

[GG12] M. Goerigk and B. Grün. The robust bus evacuation problem. Technical report, Fachbereich Mathematik, Technical University of Kaiserslautern, 2012.

[GGH13a] M. Goerigk, B. Grün, and Ph. Heßler. Branch and bound algorithms for the bus evacuation problem. Technical report, Fachbereich Mathematik, Technical University of Kaiserslautern, 2013.

[GGH13b] M. Goerigk, B. Grün, and Ph. Heßler. A branch-cut-and-price approach to the bus evacuation problem with integrated collection point and shelter decisions. Technical report, Fachbereich Mathematik, Technical University of Kaiserslautern, 2013.

[Ham01] S. A. Hamacher, H. W. Tjandra. Mathematical modelling of evacuation problems: a state of the art. In *Pedestrian and Evacuation Dynamics*, pages 227–266. Springer, Berlin, 2001.

[LLMS09] C. Liebchen, M. Lübbecke, R. H. Möhring, and S. Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In R. K. Ahuja, R.H. Möhring, and C.D. Zaroliagis, editors, *Robust and online large-scale optimization*, volume 5868 of *Lecture Note on Computer Science*, pages 1–27. Springer, 2009.

[PGAMCVT13] Francisco Pérez-Galarce, Eduardo Álvarez-Miranda, Alfredo Candia-Véjara, and Paolo Toth. On exact solutions for the minmax regret spanning tree problem. Technical report, 2013. Preprint.

[SE10] Fatemeh Sayyady and Sandra D. Eksioglu. Optimizing the use of public transit system during no-notice evacuation of urban areas. *Computers & Industrial Engineering*, 59(4):488 – 495, 2010.

[SHZ09] Rui Song, Shiwei He, and Li Zhang. Optimum transit operations during the emergency evacuations. *Journal of Transportation Systems Engineering and Information Technology*, 9(6):154 – 160, 2009.

[Soy73] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

[Sti08] S. Stiller. *Extending concepts of reliability. Network creation games, real-time scheduling, and robust optimization.* PhD thesis, TU Berlin, 2008.

[YMC09] Tao Yao, Supreet Mandala, and Byung Chung. Evacuation transportation planning under uncertainty: A robust optimization approach. *Networks and Spatial Economics*, 9(2):171–189, 2009.

[ZLZW10] Hailin Zhang, Hao Liu, Hailin Zhang, and Jian Wang. Modeling of evacuations to no-notice event by public transit system. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 480–484, 2010.