

Zone-based, Robust Flood Evacuation Planning[☆]

Sabine Büttner^a, Marc Goerigk^{b,*}

^a*Technische Universität Kaiserslautern, Kaiserslautern, Germany*

^b*University of Lancaster, Lancaster, United Kingdom*

Abstract

We consider the problem to evacuate several regions due to river flooding, where sufficient time is given to plan ahead. To ensure a smooth evacuation procedure, our model includes the decision which regions to assign to which shelter, and when evacuation orders should be issued, such that roads do not become congested.

Due to uncertainty in weather forecast, several possible scenarios are simultaneously considered in a robust optimization framework. To solve the resulting integer program, we apply a Tabu search algorithm based on decomposing the problem into better tractable subproblems. Computational experiments on random instances and an instance based on Kulmbach, Germany, data show considerable improvement compared to an MIP solver provided with a strong starting solution.

Keywords: Evacuation Planning, Flood Evacuation, Robust Optimization

1. Introduction

World-wide, flood damages have been on the rise and will likely continue to pose a further increasing risk. The floods in June 2013 are estimated to amount to 12 billion Euro in losses [Re13], while the floods in India and Pakistan during September 2014 caused over 5 billion US\$ and 665 fatalities alone [Re14]. Yearly losses within the EU are estimated to increase to over 23 billion Euro annually by 2050 [JHSF⁺14].

Due to highly sophisticated weather forecasts, floods do not hit us completely by surprise, and do not belong to the class of no-notice emergencies. Therefore, operations research has great potential to prepare for and to mitigate flood effects. Still, the literature on optimization models specifically for flood evacuation has been relatively sparse so far in comparison to other natural disasters [AG06]; for general surveys on evacuation planning and disaster management, we refer to [HT01, AG06].

[☆]Partially supported by the German Ministry of Research and Technology (BMBF), project RobEZiS, grant number 13N13198.

*Corresponding author; Address: Lancaster University Management School, Bailrigg, Lancaster, LA1 4YX, United Kingdom; Phone: +44 (0)1524 595125

Email addresses: buettner@mathematik.uni-kl.de (Sabine Büttner),
m.goerigk@lancaster.ac.uk (Marc Goerigk)

In [KYC05], a game-theoretic model for shelter location and allocation in flood evacuation planning is presented, which amounts to a bi-level optimization program that is solved using a genetic algorithm. Other papers include data uncertainty by developing robust and stochastic optimization models. In [CTC07], a two-stage program for flood evacuation is presented that includes the location of storehouses and the allocation and distribution of resources. To solve this model, sample average approximation is applied. A robust model for issuing evacuation instructions is presented in [HHH11], which extends the previous work [HHPB10]. An ant colony algorithm is applied to solve the model.

Robust optimization has been successfully applied also to other problems in disaster management, e.g., in [GDT15, GG14, KWLY11]. For general overviews on robust optimization, see [KY97, ABV09, BTGN09, GS15].

In this paper we present a model for planning an evacuation due to inland flooding, that takes the decision when to issue the evacuation orders, and the assignment of regions to shelters into account. In Section 2 we discuss this model in detail, before presenting a formulation as an integer program (IP) and showing its \mathcal{NP} -hardness in Section 2.2. Acknowledging the data uncertainty present in our problem, we discuss a robust model extension in Section 3. To solve this model, we consider a decomposition of the problem in Section 4 and a Tabu search heuristic in Section 5. Our solution algorithms are tested in computational experiments that are described in Section 6. Section 7 concludes our paper and points to further areas of research.

2. Nominal Model Definition

2.1. Model Purpose and Description

To simplify the presentation, we first introduce our flood evacuation model without data uncertainty, i.e., when weather forecasts are reliable. We then extend this formulation to several scenarios in Section 3.

Consider the following evacuation problem, which we denote as the flood evacuation problem (FEP): in foresight of a river flooding, an evacuation plan has to be determined for a certain landscape (e.g. a city) consisting of an assignment of the evacuees to one of a number of available *shelters* and, additionally, an (individual) evacuation order, i.e. a time at which the instruction to evacuate is issued. The area of consideration is divided into certain *regions* (e.g. quarters) and each of the regions itself is subdivided into several *zones* (e.g. blocks), see also Figure 1.

When allocating the evacuees to the shelters, not only the overall capacity of the shelters has to be respected but due to organizational and/or infrastructural reasons (e.g. registration upon arrival or restricted access roads), the number of people arriving at a shelter at the same time is not arbitrary but encounters a shelter depending upper bound. In addition, real world evacuation situations in which evacuation orders are given but the actual decision when to leave is left to the evacuees show that most often not all evacuees evacuate immediately: a part might leave right away but others may

leave a certain time span later than receiving the instruction. Hence, a certain *response time* can be observed.

Finally, in the situation of a river flooding, not all areas are exposed to the same risk from a chronological view. Some areas are flooded earlier than others (those lying upstream and those closer to the river.)

To incorporate all of these properties we propose the following model (in discrete time-steps): The regions within the area to be evacuated and the zones within each region as well as the set of available shelters are known. For each of the zones, there is a *latest notification time* at which the evacuation order should be issued at latest. The violation of these target times will be the criterion when determining an optimal evacuation plan. Furthermore, the total number of evacuees in a zone (the *demand*) is known but, moreover, we know the *response profile* of the zone: the total demand is split into the number of people leaving directly after receiving the evacuation order, the number leaving one time step after that, two time steps delayed, etc. This profile is given independently for each zone. This allows to model the fact that usually an evacuee living closer to the waterside will leave earlier than somebody living more inland and we assume these values to be fixed (e.g. estimated due to evacuees' behavior in earlier flooding situations).

For each of the shelters two values are given: on the one hand the total *capacity* giving the maximum number of people assigned to the same shelter, and on the other hand also an *accommodation rate* giving the maximum number of people arriving at the shelter per time step. Both capacity constraints are strict. Due to budgeting reasons, not all available shelters can be used; instead, there is a bound given on the number of shelters that may be opened.

Then, a feasible evacuation plan determines an assignment of the regions to the shelters and an evacuation order time individually for each zone such that the overall capacity of no shelter is exceeded and the number of evacuees arriving at a shelter at the same time does not exceed the accommodation rate. In particular, all evacuees from the zones within one region are assigned to the same shelter, but a shelter can host evacuees from several regions (as long as both the overall capacity restriction and – at each time – the bound on the number of arriving evacuees from all regions w.r.t. the respective evacuation orders times and the response profile is fulfilled).

The plan is considered optimal if the number of evacuees being informed (and thus evacuated) later than their respective target time is minimal.

An example for response profiles of two zones is given in Figures 2(a) and 2(b). If the shelter accommodation rate is equal to 20, we cannot give an evacuation order for both zones to start at the first timestep, as the resulting sum of profiles would exceed the accommodation rate. If we postpone the departure of the second region by three timesteps, both profiles add up as shown in Figure 2(c), which becomes feasible with respect to the accommodation rate.

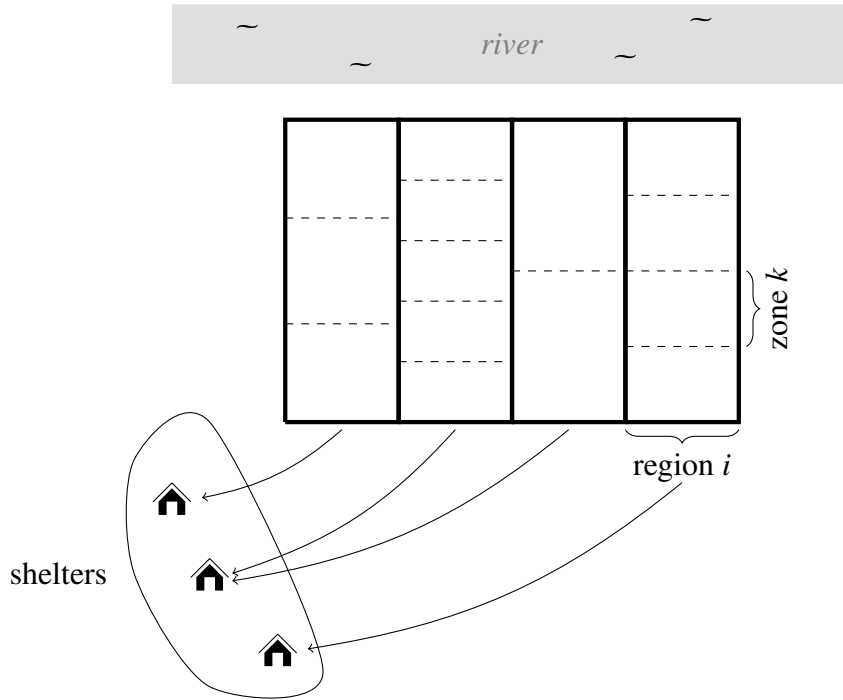


Figure 1: The zonal model (FEP).

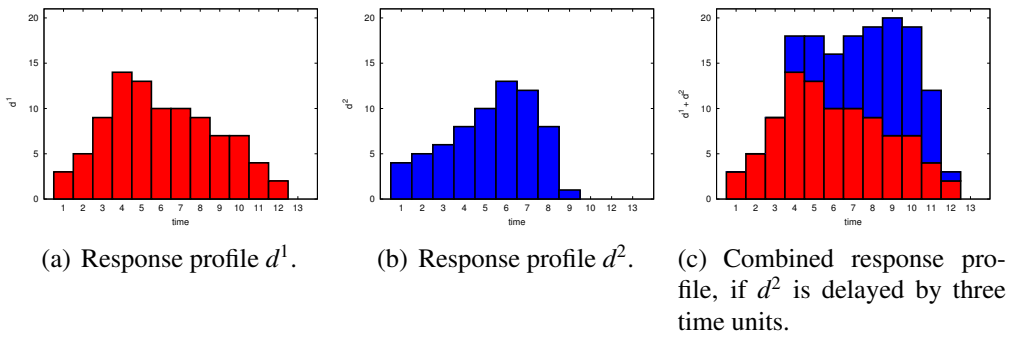


Figure 2: Example for response profiles with feasible evacuation order when the accommodation rate is 20.

2.2. IP formulation

We now describe an integer programming model for (FEP), beginning with data and variables in Section 2.2.1, followed by objectives and constraints in Section 2.2.2.

2.2.1. Data and Variables

The following sets are given in the (FEP):

- regions $\mathcal{I} = \{1, \dots, n\}$,

- *zones* $\mathcal{K}_i = \{1, \dots, k_i\}$ for each region $i \in I$,
- *shelters* $\mathcal{S} = \{1, \dots, m\}$.

For each zone $k \in \mathcal{K}_i$ we have:

- *response profile* $[d_0^{ik}, \dots, d_{t_k}^{ik}]$, where d_t^{ik} is the demand in zone k , in t periods after the evacuation order is given.
 D^{ik} is the overall demand in zone k and D_i denotes the overall demand of region i , that is $D^{ik} := \sum_{0 \leq \tau \leq t_k} d_\tau^{ik}$ and $D_i := \sum_{k \in \mathcal{K}_i} D^{ik}$,
- *desired notification time* Δ_{ik} until which the evacuation order should be issued. The desired notification time should be chosen in a way that, depending on the respective response profile, all evacuees have left the region before being endangered.

For each shelter $j \in \mathcal{S}$ we have:

- *capacity* u_j , giving the maximum number of evacuees in shelter j ,
- *accommodation rate* b_j , the maximum number of evacuees arriving at j per period.

Additionally, we have:

- *time horizon* $T \in \mathbb{N}$. We set $\mathcal{T} = \{1, \dots, T\}$.
- *maximum number of shelters* C , denoting the number of shelters from \mathcal{S} that can be used.

Finally, we assume when any evacuation order is given within the time horizon $\mathcal{T}' \subseteq \mathcal{T}$, there is sufficient time available within \mathcal{T} for the evacuation to be completed (for the simplicity of notation, we assume \mathcal{T}' to be the same for all regions and zones).

We furthermore introduce the following variables:

- $z_{ij} \in \{0, 1\} =: \mathbb{B}$ to model if the evacuees from region $i \in \mathcal{I}$ are assigned to shelter $j \in \mathcal{S}$
- $x_{ikt} \in \mathbb{B}$ to model if the evacuation order for zone $k \in \mathcal{K}_i$ in region $i \in \mathcal{I}$ is given at time $t \in \mathcal{T}'$
- $s_j \in \mathbb{B}$ to model if shelter $j \in \mathcal{S}$ is opened
- $y_{iktj} \in \mathbb{N}$ is the number of evacuees leaving zone $k \in \mathcal{K}_i$ in region $i \in \mathcal{I}$ to shelter $j \in \mathcal{S}$ at time $t \in \mathcal{T}$.

2.2.2. Integer Program

Using the above notation, we formulate the (FEP) as follows:

$$\min \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \sum_{\substack{\tau \in \mathcal{T}' \\ \tau \geq \Delta_{ik}}} (\tau - \Delta_{ik}) \cdot x_{ikt} \quad (1)$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}'} x_{ikt} = 1 \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i \quad (2)$$

$$\sum_{j \in \mathcal{S}} y_{iktj} = \sum_{\tau \leq t} d_{\tau}^{ik} x_{ik(t-\tau)} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T} \quad (3)$$

$$\sum_{j \in \mathcal{S}} z_{ij} = 1 \quad \text{for all } i \in \mathcal{I} \quad (4)$$

$$\sum_{t \in \mathcal{T}} y_{iktj} \leq z_{ij} M \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{S} \quad (5)$$

$$\sum_{i \in \mathcal{I}} z_{ij} D_i \leq u_j s_j \quad \text{for all } j \in \mathcal{S} \quad (6)$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} y_{iktj} \leq b_j s_j \quad \text{for all } j \in \mathcal{S}, t \in \mathcal{T} \quad (7)$$

$$\sum_{j \in \mathcal{S}} s_j \leq C \quad (8)$$

$$x_{ikt} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}' \quad (9)$$

$$z_{ij} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, j \in \mathcal{S} \quad (10)$$

$$s_j \in \mathbb{B} \quad \text{for all } j \in \mathcal{S} \quad (11)$$

$$y_{iktj} \in \mathbb{N} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}, j \in \mathcal{S} \quad (12)$$

The constant parameter M in constraint (5) is chosen sufficiently large, e.g. $M = \sum_i D_i$.

The objective (1) is to minimize the number of evacuees who are evacuated too late with respect to the respective notification time Δ_{ik} (weighted with the severity of the delay). Note that if the evacuation order is issued before Δ_{ik} , then the corresponding term in the objective becomes zero; for delayed evacuation orders, the penalty increases linearly.

A solution of the IP encodes a solution to the (FEP): for each zone, exactly one evacuation order is issued (2) and the number of evacuees leaving a zone are computed accordingly (3). All evacuees are assigned to one shelter (4) and, moreover, all evacuees from one region are assigned to the same shelter (5). The maximum capacity of each shelter is not exceeded (6) and, at no time, more than the allowed number of evacuees arrive at a shelter (7). Finally, Constraint (8) ensures that at most C shelters are being used.

2.3. Hardness

The (FEP) is \mathcal{NP} -hard in the strong sense as can be seen e.g. by a reduction from 3-PARTITION:

3-PARTITION

instance: $A = \{a_1, \dots, a_{3m}\}$ where $a_i \in \mathbb{Z}$, $\sum_i a_i = mB$ and $\frac{B}{4} < a_i < \frac{B}{2}$ for all i .

question: Can A be partitioned into disjoint sets A_1, \dots, A_m such that $\sum_{a \in A_j} a = B$ for all j ?

Given an instance of 3-PARTITION, we construct an instance of (FEP) as follows: there are $3m$ regions with only one zone each. The response profile of this zone in region i is $[a_i]$, i.e. there is no delayed response but all evacuees leave right after receiving the order. The target time is $\Delta = m - 1$ for all zones, s.t. the best objective value equals 0 if and only if there is a solution in which all zones receive the evacuation order within the first m time steps. Additionally there is a single shelter with capacity $3m$ and accommodation rate B . The time-horizon is sufficiently large. Note that this construction can be done in polynomial time.

Then, a solution with objective value equal to zero is equivalent to a partition of the set A into triples of weight B and, hence, solving (FEP) allows answering the 3-PARTITION problem. As 3-PARTITION is \mathcal{NP} -hard in the strong sense (cf. [GJ79]), this also holds for the (FEP).

2.4. Determining the Time Horizon

We now discuss how the parameter T can be found such that it is large enough to allow a feasible solution to (FEP); at the same time, finding the smallest possible such T improves the computational performance when solving (FEP).

To this end, let l_{ik} denote the length of the evacuation profile d^{ik} , i.e.,

$$l_{ik} = \max \{t \in \mathcal{T} : d_t^{ik} > 0\} - \min \{t \in \mathcal{T} : d_t^{ik} > 0\}.$$

For each shelter $j' \in \mathcal{S}$, we solve a subproblem that determines the maximum sum of such lengths that can be feasibly assigned to j' . Let $\bar{d}_i = \max_{k \in \mathcal{K}_i} \max_{t \in \mathcal{T}} d_t^{ik}$. For fixed j' , the optimization problem we consider is given by

$$\max \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} l_{ik} z_{ij'} \quad (13)$$

$$\text{s.t. } \sum_{j \in \mathcal{S}} z_{ij} = 1 \quad \text{for all } i \in \mathcal{I} \quad (14)$$

$$\sum_{i \in \mathcal{I}} D_i z_{ij} \leq u_j s_j \quad \text{for all } j \in \mathcal{S} \quad (15)$$

$$\bar{d}_i z_{ij} \leq b_j s_j \quad \text{for all } j \in \mathcal{S} \quad (16)$$

$$\sum_{j \in \mathcal{S}} s_j \leq C \quad (17)$$

$$z_{ij} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, j \in \mathcal{S} \quad (18)$$

$$s_j \in \mathbb{B} \quad \text{for all } j \in \mathcal{S} \quad (19)$$

As before, Constraints (14) ensure that every region is assigned to one shelter, while Constraints (15) and (16) model that both shelter capacity and shelter accommodation rate are respected. Constraint (17) bounds the number of shelters that can be opened by C . The objective function (13) aims at finding the worst-case over all assignments to j' . Solving (13–19) for all shelters, and taking the maximum of all optimal values, then gives an upper bound on the value of T .

3. The Robust Problem

We now extend the nominal (FEP) model from Section 2.2 to include data uncertainty. To this end, we assume the following setting:

As the severity of a flood cannot be forecasted with complete certainty, there are different scenarios possible, which should all be taken into account during the planning step. The set of possible scenarios is called the uncertainty set \mathcal{U} . The flood severity affects both the evacuation profile d and the desired order issue time Δ , such that we set

$$\mathcal{U} = \{(d^1, \Delta^1), \dots, (d^N, \Delta^N)\}$$

The assignment of regions to shelters is a question of long-term planning, and should be done well in advance of the actual evacuation (e.g., shelters must be equipped for the right number of evacuees, endangered regions must be informed which path to use during an evacuation, etc.). Hence, the decision variables z_{ij} and s_j must be the same for all scenarios. However, the actual time of notification for the evacuation order can be adapted to the circumstances; hence, variables x_{ikt} and y_{iktj} may be decided upon once the scenario is known to the planner. We therefore adapt a two-stage setting, and use sets of variables x^ξ and y^ξ for each scenario.

We now present an IP formulation using the resulting robust model, which we call (RFEP). We set $\mathcal{N} := \{1, \dots, N\}$.

$$\min \zeta \tag{20}$$

$$\text{s.t. } \zeta \geq \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \sum_{\substack{\tau \in \mathcal{T}' \\ \tau \geq \Delta_{ik}^\xi}} (\tau - \Delta_{ik}^\xi) \cdot x_{ikt}^\xi \quad \text{for all } \xi \in \mathcal{N} \tag{21}$$

$$\sum_{t \in \mathcal{T}'} x_{ikt}^\xi = 1 \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, \xi \in \mathcal{N} \tag{22}$$

$$\sum_{j \in \mathcal{S}} y_{iktj}^\xi = \sum_{\tau \leq t} d_\tau^{ik\xi} x_{ik(t-\tau)}^\xi \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}, \xi \in \mathcal{N} \tag{23}$$

$$\sum_{j \in \mathcal{S}} z_{ij} = 1 \quad \text{for all } i \in \mathcal{I} \tag{24}$$

$$\sum_{t \in \mathcal{T}} y_{iktj}^\xi \leq M z_{ij} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{S}, \xi \in \mathcal{N} \tag{25}$$

$$\sum_{i \in \mathcal{I}} z_{ij} D_i \leq u_j s_j \quad \text{for all } j \in \mathcal{S} \tag{26}$$

$$\sum_{i \in \mathcal{I}} \sum_k y_{iktj}^\xi \leq b_j s_j \quad \text{for all } j \in \mathcal{S}, t \in \mathcal{T}, \xi \in \mathcal{N} \quad (27)$$

$$\sum_{j \in \mathcal{S}} s_j \leq C \quad (28)$$

$$x_{ikt}^\xi \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}, \xi \in \mathcal{N} \quad (29)$$

$$z_{ij} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, j \in \mathcal{S} \quad (30)$$

$$s_j \in \mathbb{B} \quad \text{for all } j \in \mathcal{S} \quad (31)$$

$$y_{iktj}^\xi \in \mathbb{N} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}, j \in \mathcal{S}, \xi \in \mathcal{N} \quad (32)$$

$$\zeta \geq 0 \quad (33)$$

The new variable ζ is used to determine the worst-case over all scenarios with the help of Constraints (21). The remaining Constraints (22–33) are a direct extension of Constraints (2–12). Note that (RFEP) is also \mathcal{NP} -hard, containing (FEP) as a special case.

4. Problem Decomposition

To better access model (RFEP) using heuristic solution approaches, we decompose the problem into two subproblems: Firstly, to find a feasible assignment of regions to shelters; and secondly, to find a good set of evacuation order times for all zones assigned to a fixed shelter and under a fixed scenario.

4.1. Finding a Feasible Assignment

To find a feasible assignment, we solve the following problem (similar to the problem described in Section 2.4). Let $\bar{d}'_i := \max_{k \in \mathcal{K}_i} \max_{t \in \mathcal{T}} \max_{\xi \in \mathcal{N}} d_t^{ik\xi}$ denote the maximum number of evacuees starting at any zone at any time in any scenario. To help with the subsequent determination of evacuation order times, we choose a subset of shelters that maximize the total accommodation rate, using the following integer program:

$$\max \sum_{j \in \mathcal{S}} b_j s_j \quad (34)$$

$$\text{s.t. } \sum_{j \in \mathcal{S}} z_{ij} = 1 \quad \text{for all } i \in \mathcal{I} \quad (35)$$

$$\sum_{i \in \mathcal{I}} D_i z_{ij} \leq u_j s_j \quad \text{for all } j \in \mathcal{S} \quad (36)$$

$$\bar{d}'_i z_{ij} \leq b_j s_j \quad \text{for all } j \in \mathcal{S} \quad (37)$$

$$\sum_{j \in \mathcal{S}} s_j \leq C \quad (38)$$

$$z_{ij} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, j \in \mathcal{S} \quad (39)$$

$$s_j \in \mathbb{B} \quad \text{for all } j \in \mathcal{S} \quad (40)$$

4.2. Evaluating a Feasible Assignment

Let some feasible assignment of regions to shelters be given, i.e., values for z and s that can be extended to a feasible solution for (RFEP).

For every shelter and every scenario, we check the best possible departure times of all assigned regions. Let $I(j)$ be the regions assigned to any shelter $j \in \mathcal{S}$ in this solution, and set $n(j) := |I(j)|$. Finding evacuation order times that minimize the total delay in scenario $\xi \in \mathcal{U}$ for shelter j (i.e., the process described in Figure 2), which we call (EV) in the following, is then modeled via the following IP:

$$\min \sum_{i \in I(j)} \sum_{k \in \mathcal{K}_i} \sum_{\Delta_{ik} \leq \tau \leq T} (\tau - \Delta_{ik}^{\xi}) \cdot x_{ikt} \quad (41)$$

$$\text{s.t.} \quad \sum_t x_{ikt} = 1 \quad \text{for all } i \in I(j), k \in \mathcal{K}_i \quad (42)$$

$$y_{ikt} = \sum_{\tau \leq t} d_{\tau}^{ik\xi} x_{ik(t-\tau)} \quad \text{for all } i \in I(j), k \in \mathcal{K}_i, t \in \mathcal{T} \quad (43)$$

$$\sum_{i \in I(j)} \sum_{k \in \mathcal{K}_i} y_{ikt} \leq b_j \quad \text{for all } t \in \mathcal{T} \quad (44)$$

$$x_{ikt} \in \{0, 1\} \quad \text{for all } i \in I(j), k \in \mathcal{K}_i, t \in \mathcal{T} \quad (45)$$

$$y_{ikt} \in \mathbb{N} \quad \text{for all } i \in I(j), k \in \mathcal{K}_i, t \in \mathcal{T} \quad (46)$$

Problem (EV) of evaluating a feasible assignment is in itself already strongly \mathcal{NP} -hard, as the same proof from Section 2.3 can be applied.

5. Heuristic Solution Algorithms

We now introduce heuristic methods for (RFEP) based on the problem decomposition presented in Section 4.

5.1. Heuristic Evaluation of an Assignment

The problem of finding a feasible assignment as described in Section 4.1 is comparably small and is computationally easy to handle; therefore, we focus here on the subsequent evaluation problem (EV). As problem (EV) will need to be solved frequently by an algorithm that examines different possible assignments of regions to shelters, we consider a procedure that takes little computation time.

To this end, let us assume all zones included in an instance of (EV) are given in a list determining their priority. We begin with the first element from the list and include it in the current solution as early as possible, i.e., at time zero. We then proceed with the subsequent zones from the list and always fix the evacuation order time as early as possible, such that the accommodation rate of the shelter is never exceeded.

Additionally, we perform a local search over the current list by testing if exchanging any two zones improves the resulting objective value until no further improvement is reached.

One natural possibility to determine such a starting list of zones is to sort them by their desired notification time Δ . Additionally, we use randomly ordered lists to restart the local search process once a local minimum has been found. The number of such restarts is a tunable parameter, where more restarts may result in a better objective value for (EV), but also longer computation times.

Note that there are instances of (EV), where no optimal solution has the structure of solutions that we consider in this heuristic. It therefore does not even have an approximation guarantee. As an example, consider an instance with $T = 5$ and $b = 8$. There are three zones that need to be scheduled, with $d^1 = [7, 6, 4, 2]$, and $d^2 = d^3 = [1, 2, 3]$. We further have $\Delta^1 = 0$ and $\Delta^2 = \Delta^3 = 1$. To obtain a solution with objective value equal to zero, the first zone needs to depart at time zero. Now, if we schedule any of the other zones as early as possible, the other zone can only depart at time two, leading to a delay of one overall. If, however, the other two zones depart at time one, there is no delay at all. This example shows that it may be impossible for the proposed heuristic to find an optimal solution to the evaluation problem; however, the reduced search space gives as speedup for the evaluation as a trade-off.

5.2. A Tabu Search Algorithm

We now combine the heuristic solution of (EV) with an algorithm to inspect different assignments of regions to shelters, including the choice which of the shelters to open. We therefore operate on both levels of the problem decomposition described in Section 4, where computation time spent on finding new assignments corresponds to the exploration part of the algorithm, and computation time spent of finding a good evaluation to a given assignment to the exploitation side.

Given a feasible assignment, we consider the following local search moves:

- Move all regions assigned to one shelter to a new shelter that was previously closed.
- Take one region assigned to one shelter and another region assigned to another shelter, and swap their assignments.
- Take a single region assigned to one shelter and assign it to another shelter (possibly opening a shelter in the process).

We keep a fixed-size, first-in-first-out Tabu list of previous assignments, that may not be visited again. To help the exploration process, we also allow infeasible assignments, and use three penalty parameters: one for exceeding the allowed number of shelters C , one for exceeding the shelter capacity u_j , and one for exceeding the accommodation rate b_j . To this end, we adapt our heuristic procedure for (EV) such that if no feasible set of evacuation order times exists, we choose one that minimizes the sum of violations of the accommodation rate. The objective value used for the Tabu search is then given as a combination of the actual problem objective value, and an additional weighted sum of penalties.

Whenever the current solution of our Tabu search is feasible, we reduce the penalties by a small, random amount (such that infeasible solutions become more attractive); if the current solution is infeasible, these penalties are increased (such that feasible solutions become more attractive). Whenever a solution is encountered that is feasible and better than the current best, it is immediately chosen; if this is not the case, the best solution according to the modified objective that is not in the Tabu list is chosen in each iteration. We randomly shuffle all feasible moves so that they are not always encountered in the same order.

If there is no solution in the neighborhood that is not also in the Tabu list, we reset the search. To this end, the best solution found so far is restored, the Tabu list is emptied, and all penalties are set to their original range.

To guide the search from exploration to exploitation, we use different numbers of reshuffles in our evaluation heuristic. In each iteration, this number is slightly increased, such that the evaluation of the current neighborhood takes more time, but potentially better solutions can be found.

6. Experiments

6.1. Setup and Environment

We generate two sets of instances P and Q , such that instances of type P allow a large number of feasible assignments (requiring an algorithm with good exploration properties), and instances of type Q allow only few feasible assignments (making exploitation properties more relevant).

Both types of instances were generated in the following way. Given the number of regions n , the number of shelters m , the number of scenarios N , and the number of shelters that can be opened C , we generate all other parameters. We uniformly randomly choose $u_j \in \{300, \dots, 800\}$, $D_i \in \{100, \dots, 400\}$, and $k_i \in \{1, \dots, 5\}$ for all i and j . We then distribute the number of people D_i of region i randomly over the k_i zones within this region. For each scenario, we then further generate d by randomly distributing the number of people within each zone over the first 20 timesteps, such that the response profiles are different in each scenario, but the number of evacuees is the same. We check if a feasible assignment exists; if this is not the case, all parameters are generated again, until the instance becomes feasible. Finally, we choose Δ randomly from $\{0, \dots, 4\}$ for each zone and scenario.

We show the parameter choice for n , m , and C in Table 1. We generated all instances with two and with five scenarios. For each parameter set, we generated ten instances (i.e., a total of 340 instances). In the following, we denote by P_n or Q_n the sets of instances with n regions.

Each instance is solved using its MIP formulation and by our Tabu search heuristic. The MIP solver (Cplex) is provided with a feasible starting solution (otherwise, preliminary experiments showed that in most larger instances, not even a feasible solution could be found). To this end, we use the decomposed MIPs described in Section 4 and

n	P		Q	
	m	C	m	C
4	6	2	6	2
5	9	3	6	2
6	9	3	6	2
7	12	4	9	3
8	12	4	9	3
9	15	5	9	3
10	15	5	12	4
11	18	6	12	4
12	18	6	12	4

Table 1: Instance parameters.

allow at most half of the total timelimit to be devoted to solving the evaluation problem. For better comparability, the MIPEmphasis Cplex parameter is chosen so that lower bounds are neglected and more computation time is spent on finding feasible solutions. For our Tabu search, only the initial assignment is found using a MIP solver; all evaluations are then carried out using our heuristic algorithm. Each algorithm was given a timelimit of 10 minutes.

Additionally, we used the MIP solver with an emphasis on lower bounds and a computation time of 60 minutes for all instances to find strong lower bounds. Because some lower bounds were equal to zero, we normalized all objective values with an offset such that the lower bound is always equal to 100.

All experiments were conducted on a computer with a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 20MB cache, using one core per algorithm. To solve MIPs, CPLEX v.12.6 ([IBM13]) was used. The Tabu search was run five times for every instance to account for its randomness.

6.2. Results

We present the average gaps over the ten instances of each size in Table 2 for the P instances, and in Table 3 for the Q instances. The gap is computed as $(UB - LB)/UB$, where UB denotes the respective objective value, and LB the lower bound. In column “IP”, we show the gap when using Cplex, and show the gap for our Tabu search algorithm in columns “Tab-Min” (the average gap of the best results over the five runs per instance), “Tab-Av” (the average gap over all runs), and “Tab-Max” (the average gap of the worst result of the five runs). All values are in percent.

For all P instances, except for P_4 with $N = 5$, even Tab-Max considerably outperforms the IP gap. This changes for the Q instances (Table 3), where the Tabu search still outperforms IP in most cases, but not as distinctly. Even for small instances, optimality cannot be proved in most cases, which may largely be due to weak lower bounds.

Inst	$N = 2$				$N = 5$			
	IP	Tab-Min	Tab-Av	Tab-Max	IP	Tab-Min	Tab-Av	Tab-Max
P_4	24.68	19.83	19.83	19.83	10.91	11.97	11.99	12.04
P_5	20.76	15.46	15.53	15.81	30.15	22.23	22.42	22.73
P_6	25.49	21.42	21.59	21.90	33.63	25.54	26.59	27.92
P_7	25.95	20.33	20.61	20.96	32.42	26.88	27.15	27.87
P_8	36.97	30.56	31.15	31.60	45.66	33.68	35.45	37.13
P_9	45.38	37.06	38.60	40.99	30.66	22.05	22.40	22.99
P_{10}	35.36	29.64	30.98	32.50	37.04	26.95	28.57	30.73
P_{11}	30.30	21.29	22.88	23.99	40.24	30.79	31.96	33.41
P_{12}	35.53	19.54	21.32	24.60	40.55	32.38	34.15	36.41

Table 2: Average gap for P instances.

Inst	$N = 2$				$N = 5$			
	IP	Tab-Min	Tab-Av	Tab-Max	IP	Tab-Min	Tab-Av	Tab-Max
Q_4	24.68	19.83	19.83	19.83	10.91	11.97	11.99	12.04
Q_5	23.16	19.55	20.11	21.00	29.45	31.16	31.23	31.38
Q_6	37.45	36.48	36.64	37.02	25.38	24.23	24.75	25.02
Q_7	38.96	32.91	33.77	34.87	41.58	40.08	40.52	40.72
Q_8	40.72	36.77	37.46	37.94	47.86	43.70	46.09	46.90
Q_9	37.58	35.67	36.34	37.07	52.62	49.96	50.73	51.48
Q_{10}	41.52	40.46	41.10	41.76	41.20	35.80	37.06	38.34
Q_{11}	48.38	45.17	46.14	46.79	43.24	42.73	43.57	44.36
Q_{12}	52.35	51.15	52.39	53.05	50.66	49.65	50.83	52.21

Table 3: Average gap for Q instances.

To understand these results in more depth, we show the average difference in objective gap between the starting solution and the final solution of each algorithm in Tables 4 and 5.

Inst	$N = 2$		$N = 5$	
	IP	Tab	IP	Tab
P_4	2.99	10.03	2.31	4.86
P_5	3.97	10.41	1.30	11.02
P_6	1.97	8.63	0.60	9.52
P_7	6.66	14.48	0.19	9.31
P_8	1.45	11.06	0.00	12.55
P_9	0.00	9.21	0.38	11.62
P_{10}	0.00	8.29	0.33	11.11
P_{11}	1.67	11.40	0.00	10.38
P_{12}	1.12	18.71	0.00	10.08

Table 4: Average difference of gap between starting solution and final solution for P instances.

Inst	$N = 2$		$N = 5$	
	IP	Tab	IP	Tab
Q_4	2.99	10.03	2.31	4.86
Q_5	0.16	6.19	0.48	0.23
Q_6	0.00	3.30	0.00	3.36
Q_7	0.34	7.03	0.00	3.06
Q_8	0.05	5.62	0.00	3.72
Q_9	0.00	3.76	0.00	2.66
Q_{10}	1.30	3.76	0.00	6.23
Q_{11}	0.00	3.94	0.00	3.21
Q_{12}	0.00	2.92	0.00	2.43

Table 5: Average difference of gap between starting solution and final solution for Q instances.

We see that both the MIP solver and the Tabu search are less able to improve their respective starting solutions for instances of type Q than for instances of type P . Also, the Tabu search is able to improve upon its starting solution considerably more than the MIP solver; in fact, the MIP solver does not improve the starting solution at all in the vast majority of cases. This means that the gaps for IP shown in Table 3 for $N = 5$ are due to a good exploitation of the starting solution, without any exploration at all.

6.3. Kulmbach Instance

We now apply our model and algorithms to realistic data based on the Kulmbach district in north Bavaria, Germany. A map of the region can be seen in Figure 3, with keys to placenames in Tables 6 and 7. Within the Kulmbach district, the Red and White Main rivers flow together to become the Main river, and within the recent past, the region was troubled by river floodings.

We assume that some of the people based in the subdistricts colored in red (Latin letters) need to leave their homes and make way for a shelter based in one of the other subdistricts (Greek letters). The numbers of evacuees and the numbers of zones within each region are shown in Table 6; the shelter capacity (which is based on the number of inhabitants within the region) and their accommodation rate are shown in Table 7. Of the eight shelters, six may be used.

¹Based on graphic by Hagar66 under CC-BY 2.0 license, Wikimedia commons.

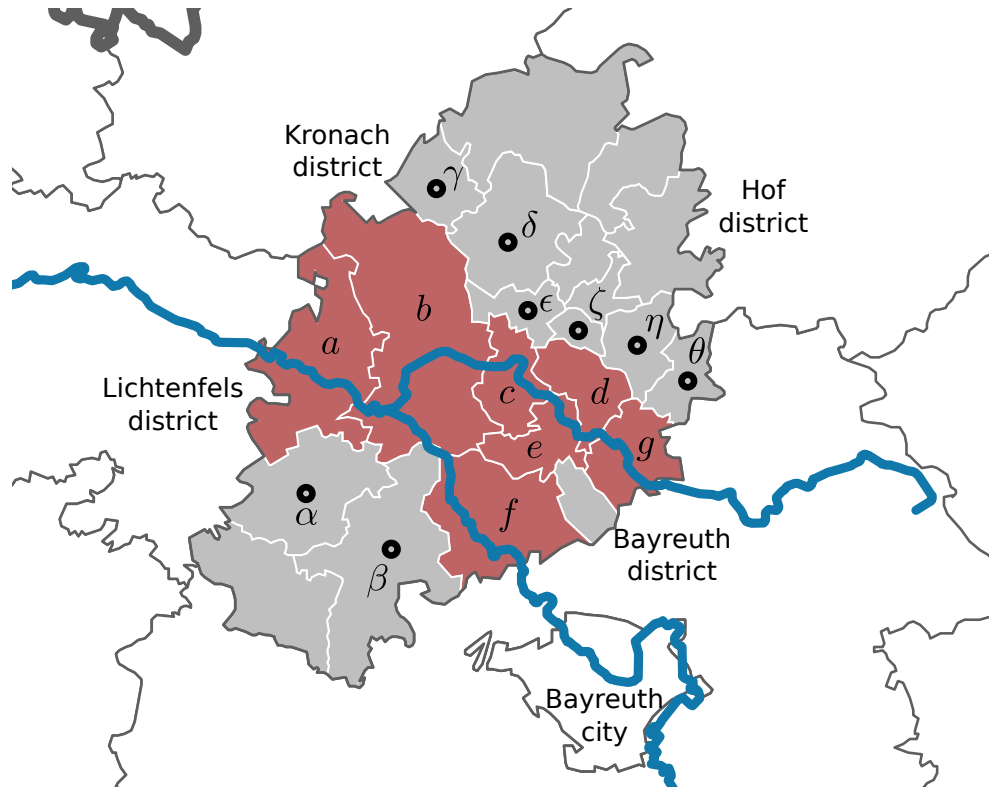


Figure 3: Kulmbach district¹

Letter	Name	Population	Evacuees	Zones
<i>a</i>	Mainleus	6,449	800	3
<i>b</i>	Kulmbach (city)	25,985	1,000	4
<i>c</i>	Ködnitz	1,583	250	2
<i>d</i>	Neuenmarkt	3,040	400	1
<i>e</i>	Trebgast	1,603	350	1
<i>f</i>	Neudrossenfeld	3,807	700	3
<i>g</i>	Himmelkron	3,490	600	2

Table 6: Regions of Kulmbach district

Letter	Name	Capacity	Acc. rate
α	Kasendorf	800	50
β	Thurnau	1,400	60
γ	Rugendorf	350	40
δ	Stadtsteinach	1,100	60
ϵ	Untersteinach	600	80
ζ	Ludwigschorgast	350	40
η	Wirsberg	600	50
θ	Markschorgast	450	30

Table 7: Sheltering regions of Kulmbach district

We generate three scenarios, which correspond to different levels of evacuation urgency: In the first scenario, profiles d are chosen such that evacuees leave early and Δ is small; in the second scenario, evacuees leave late and Δ is larger; the third scenario is between those two.

Both the IP model with Cplex and our Tabu search algorithm find the same solution within one minute of computation time, which can be shown to be optimal by running a subsequent IP with increased timelimit (as in the previous experiments).

The objective value of this solution is 49, which is reached in the first scenario (the objective values for the other scenarios are 11 and 14, respectively). Region a is assigned to shelter α , regions c and f are assigned to shelter β , region b to shelter δ , region g to shelter ϵ , region e to shelter ζ , and region d to shelter η .

7. Conclusion and Extensions

In this paper we considered the problem of coordinating the evacuation of several regions due to river flooding. We developed a model that includes for the first time several unique features of such an evacuation: The timing of evacuation orders with departure profiles is taken into account, as well as the choosing a set of shelters such that all evacuees can be accommodated without delays or traffic jams. By including several scenarios for different flooding outcomes, we take forecasting uncertainty into account via a robust optimization model.

This model decomposes into two subproblems: Assigning regions to shelters, and determining departure times for each zone. We developed a Tabu search heuristic that aims at conciliating exploration and exploitation in the interplay between these two problem levels. Our computation results show that we considerably outperform a MIP solver in terms of exploration, even if it is provided with a strong starting solution.

We now discuss possible extensions.

The first extension is to further include regional dependencies; as an example, it might not be possible to assign a region to any shelter. This can be easily facilitated by forbidding such assignments in the IP formulation. Furthermore, it is possible to in-

clude different driving times from regions to shelters: To this end, we can equivalently assume values for Δ that depend on the assignment variables z , which in turn can be linearized in the objective function, as x is a binary variable.

Additionally to the target evacuation time Δ for each zone k in any region i there might also be a *clearance time* ρ_{ik} specifying the point in time when evacuees from zone k in region i can return to their homes safely. Then, an evacuation plan must include not only a time for the zones to evacuate but also a time at which the evacuees from a zone who are already at a shelter receive the *clear order* to return home. As evacuees usually want to be back home as soon as possible we assume no delay for this but propose an immediate response., i.e. all D^{ik} evacuees actually leave at the issued time. One essential change by this additional option is that the shelters can be utilized more efficiently: returning evacuees can vacate spaces for new arrivals, i.e. the total capacity has to be respected individually at all times.

We model this in the IP-formulation of Section 2.2 by including the following sets of constraints:

$$\sum_t r_{ikt} = 1 \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i \quad (47)$$

$$\sum_{\tau < \rho_{ik}} r_{ik\tau} = 0 \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i \quad (48)$$

$$\sum_{\tau \leq t - t_k} x_{ik\tau} \geq \sum_{\tau \leq t_k} r_{ik\tau} = 0 \quad \text{for all } j \in \mathcal{S}, t \in \mathcal{T} \quad (49)$$

$$r_{ikt} \in \mathbb{B} \quad \text{for all } i \in \mathcal{I}, k \in \mathcal{K}_i, t \in \mathcal{T}. \quad (50)$$

The capacity constraints (6) are replaced by

$$\sum_{\tau \leq t} \sum_i \sum_k (y_{ik\tau j} - r_{ik\tau} d_{ik}) \leq u_j \quad \text{for all } j \in \mathcal{S}, t \in \mathcal{T}. \quad (6')$$

The binary variable r_{ikt} indicates the clearance order:

$$r_{ikt} = \begin{cases} 1, & \text{if the evacuees from zone } k \text{ in region } i \text{ receive clearance order at time } t, \\ 0, & \text{else.} \end{cases}$$

Each zone k receives exactly one clear order (47). The times of the orders is valid as it is, on the one hand, determined not before the given earliest clearance time ρ_{ik} (48) and, on the other, not before all evacuees from the zone have actually arrived at the shelter (49). (Note, that we view ρ_{ik} as a hard constraint. In contrast, target time Δ_{ik} is a soft bound and can be violated in turn for an increase in the objective value. Hence, without constraints (48) an evacuation plan could in fact schedule a clearance order earlier than the actual evacuation order. We do not want to include the option that some evacuees are in fact not informed at all.) Finally, the capacity constraints are replaced by the requirement (6') that the capacity is respected at all times by the number of people who are at the shelter in this very moment, i.e. the difference between the evacuees who have arrived and those having left.

References

- [ABV09] H. Aissi, C. Bazgan, and D. Vanderpooten. Minmax and minmax regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427 – 438, 2009.
- [AG06] N. Altay and W.G. Green III. OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475 – 493, 2006.
- [BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, Princeton and Oxford, 2009.
- [CTC07] M.-S. Chang, Y.-L. Tseng, and J.-W. Chen. A scenario planning approach for the flood emergency logistics preparation problem under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 43(6):737 – 754, 2007. Challenges of Emergency Logistics Management.
- [GDT15] M. Goerigk, K. Deghdak, and V. TKindt. A two-stage robustness approach to evacuation planning with buses. *Transportation Research Part B: Methodological*, 78:66 – 82, 2015.
- [GG14] M. Goerigk and B. Grün. A robust bus evacuation model with delayed scenario information. *OR Spectrum*, 36(4):923–948, 2014.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to NP-completeness*, 1979.
- [GS15] M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. In L. Kliemann and P. Sanders, editors, *Algorithm Engineering: Selected Results and Surveys*, volume 9220 of *Lecture Notes in Computer Science*, page 0. Springer Berlin / Heidelberg, 2015.
- [HHH11] O. Huibregtse, A. Hegyi, and S. Hoogendoorn. Robust optimization of evacuation instructions, applied to capacity, hazard pattern, demand, and compliance uncertainty. In *Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on*, pages 335–340, April 2011.
- [HHPB10] O. L. Huibregtse, S. P. Hoogendoorn, A. J. Pel, and M. C. J. Bliemer. A generic method to optimize instructions for the control of evacuations. In *12th IFAC symposium on transportation systems, Redondo Beach, Sept. 2009*. IFAC, 2010.
- [HT01] H.W. Hamacher and S.A. Tjandra. Mathematical modeling of evacuation problems: A state of the art. In *Pedestrian and Evacuation Dynamics (Schreckenberg, M. and Sharma, S. D. eds)*, 1964:227–266, 2001.

- [IBM13] IBM. *IBM ILOG CPLEX 12.6 User's Manual*, 2013.
- [JHSF⁺14] B. Jongman, S. Hochrainer-Stigler, L. Feyen, J.C.J.H. Aerts, R. Mechler, W.J.W. Botzen, L.M. Bouwer, G. Pflug, R. Rojas, and P.J. Ward. Increasing stress on disaster-risk finance due to large floods. *Nature Climate Change*, 4(4):264–268, 2014.
- [KWLY11] A. Kulshrestha, D. Wu, Y. Lou, and Y. Yin. Robust shelter locations for evacuation planning with demand uncertainty. *Journal of Transportation Safety & Security*, 3(4):272–288, 2011.
- [KY97] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.
- [KYC05] S. Kongsomsaksakul, C. Yang, and A. Chen. Shelter location-allocation model for flood evacuation planning. *Journal of the Eastern Asia Society for Transportation Studies*, 6:4237–4252, 2005.
- [Re13] Munich Re. Munich reinsurance company geo risks research. *NatCat-SERVICE Database*, 2013.
- [Re14] Munich Re. Munich reinsurance company geo risks research. *NatCat-SERVICE Database*, 2014.