Robust Flows with Losses and Improvability in Evacuation Planning [*]

Marc Goerigk[†1] and Ismaila Abderhamane Ndiaye[2]

[1] Department of Management Science, Lancaster University, United Kingdom
[2] Université François-Rabelais de Tours, LI EA 6300, OC ERL CNRS 6305, Tours, France

Abstract

We consider a network flow problem, where the outgoing flow is reduced by a certain percentage in each node. Given a maximum amount of flow that can leave the source node, the aim is to find a solution that maximizes the amount of flow which arrives at the sink.

Starting from this basic model, we include two new, additional aspects: On the one hand, we are able to reduce the loss at some of the nodes; on the other hand, the exact loss values are not known, but may come from a discrete uncertainty set of exponential size.

Applications for problems of this type can be found in evacuation planning, where one would like to improve the safety of nodes such that the number of evacuees reaching safety is maximized.

We formulate the resulting robust flow problem with losses and improvability as a two-stage mixed-integer program with uncertain recourse for finitely many scenarios, and present an iterative scenario-generation procedure that avoids the inclusion of all scenarios from the beginning as well as several heuristic solution methods. In a computational study using both randomly generated instances and realistic data based on the city of Nice, France, we compare our solution algorithms.

Keywords: network flow; flow with losses; robust optimization; adjustable robustness; random recourse; network design

1 Introduction

Network flow with losses (and also gains) is a well-known and fruitful object of study, see, e.g., [Old01, Rad98, Way99]. Applications can typically be found, e.g., in telecommunication networks, electrical networks, exchange markets, machine loading, lot-sizing or the Boolean satisfiability problem.

Also the field of evacuation planning has seen rising interest in the application of operations research models to help the decision maker assessing a

[*] 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).

[†] Corresponding author. Email: m.goerigk@lancaster.ac.uk

critical situation, and making the right choices to potentially safe lives, see, e.g., [OMH09, LKN11]. For general overviews, we refer to [HT01, AG06]. Also, network flows are a standard modeling technique in the field, see, e.g., [CFS82, Yam96, CHT88].

In this paper, we extend the concept of flows with losses to also include improvability (i.e., the amount of loss can be reduced at a limited number of nodes) as well as robustness (i.e., the exact amount of loss is not known exactly). Both model extensions are motivated by applications in evacuation planning.

As an example, we consider the situation that several evacuees need to leave an endangered region on foot (e.g., after an earthquake or a flooding strikes an urban area). Depending on the path they choose, they face different estimated degrees of dangerousness, which lead to the potential death of evacuees. Such a situation can be captured as a network flow with losses, where the number of evacuees reaching safety is to be maximized (see [NNL$^+$14]).

From a short-term perspective, security forces can be used to reduce the risk in the network (e.g., by extinguishing fires with the help of airplanes, or by removing debris) during an evacuation. From a long-term perspective, the structural safety of an endangered area can be improved (e.g., by stabilizing buildings with a relatively high probability of collapse during an earthquake), which will also result in a reduced risk value during an emergency.

Furthermore, as risk values are only an estimation, they are considered as being uncertain. We present a robust optimization approach with network improvability to include both these points.

In the following, we present some further literature in which related aspects are analyzed.

The basic idea of the well-known "Contraflow"-setting is to make better use of the given infrastructure in the case of an evacuation. As an example, if a highway has two lanes entering the endangered area, and two lanes leaving the endangered area, it makes sense to reverse at least one of the two entering lanes to facilitate the outgoing flow. The contraflow problem has been considered by [XLW10, XT11] and many others. Similar to our setting, the lane reversal may be interpreted as the distribution of improvements in the network, where the number of such improvements is bounded. However, in our setting, improvements are not on arc capacities, but on vertex safety instead.

There are several papers considering network improvement problems (such as [SK98, KMN$^+$98, DNW04, OZ07, DLG11, LM13, CLZ06]). However, the problem of improving vertex safety has not been considered yet.

Regarding the field of robust optimization in general as a means to handle optimization problems affected by uncertainty, we refer to the surveys [KY97, ABV09, BS04, BS03, BTGN09] and [GS15]. In our setting, we follow a two-stage adjustable robust approach (see [BTGGN04]). As the resulting optimization problem is too large to be solved directly, we employ an iterative scenario-generation method (see also [ACF$^+$13, BCP14, GLMR14, ZZ13]).

Contributions and overview. In Section 2, we introduce the nominal (i.e., non-uncertain) max flow problem with losses and improvements, which we use to model pedestrian movements during an evacuation. This model is extended to include uncertainty in Section 3, where we also present an iterative solution algorithm. As already the nominal problem is NP-hard, we also consider heuristic solution approaches in Section 4, and compare these algorithms in a computational study in Section 5. Section 6 concludes the paper and points out further research directions.

2 Flow with Losses and Improvability

2.1 Flow with Losses

Let a directed graph $G = (V, A)$ be given. We start with considering the maximum flow problem, where in each vertex $i \in V$, the flow leaving node i is multiplied with a fixed factor $p_i \in [0, 1]$. Given a bound on the value of outflow of a source, the *flow with losses* (FL) problem is to find the largest possible amount of flow entering a sink. The restriction to a single sink and a single source is without loss of generality, as multiple sources and sinks can be collected to a super-source and -sink, respectively.

The problem can be easily modeled as a linear program using the variables f_{ij} to denote the flow along arc $(i, j) \in A$. Let $s \in V$ denote the source node, and $t \in V$ the sink node. We write $V' := V \setminus \{s, t\}$. Furthermore, let $S \in \mathbb{R}^+$ denote the maximum amount of flow leaving the source node, and let the capacity of each arc $(i, j) \in A$ be denoted as $u_{ij} \in \mathbb{R}^+$.

The flow with losses problem can then be formulated as:

$$\begin{aligned}
\text{(FL)} \quad \max \quad & \sum_{(i,t)\in A} f_{it} && && (1)\\
\text{s.t.} \quad & \sum_{(s,i)\in A} f_{si} \le S && && (2)\\
& p_j \sum_{(i,j)\in A} f_{ij} = \sum_{(j,q)\in A} f_{jq} && \forall j \in V' && (3)\\
& 0 \le f_{ij} \le u_{ij} && \forall (i,j) \in A && (4)
\end{aligned}$$

The objective function (1) is to maximize the amount of flow entering the sink t. Constraint (2) ensures that at most S evacuees leave the source s, while Constraints (3) are a modification of the usual flow constraints, capturing the relative loss of flow in each node. Finally, Constraints (4) ensure that arc capacities u_{ij} are respected.

In evacuation planning, the model can be used to calculate the percentage of pedestrian evacuees that could reach safe places, starting from an endangered region. Nodes can correspond to street crossings or important points along a street, and arcs to streets or street sections connecting these points. Note that this model follows a *macroscopic* point of view, i.e., individual or selfish behavior of evacuees is ignored. Thus, a resulting optimal solution may be considered as an upper bound on the actual percentage of evacuees reaching safe places, which is only attainable if every evacuee acted according to the system optimum. Typically, macroscopic optimization results are complemented with microscopic simulation results to give a more realistic description of an emergency situation (see [HHK$^+$11]).

2.2 Flow with Losses and Improvability

We now extend the simple model FL to include possible improvements on the vertices. We assume that along with the loss p_i at every node $i \in V'$, we are also given an amount of possible improvement $\hat{p}_i$, where $p_i + \hat{p}_i \leq 1$. The improvement of any node i has costs c_i, and there is a budget B on the total possible improvement costs. We denote the problem as FLI (flow with losses and improvements).

To model FLI as a mathematical program, we start with the following non-linear model:

$$
\begin{aligned}
\max \quad & \sum_{(i,t)\in A} f_{it} && && (5)\\
\text{s.t.} \quad & \sum_{(s,i)\in A} f_{si} \leq S && && (6)\\
& \sum_{j\in V'} c_j z_j \leq B && && (7)\\
& (p_j + \hat{p}_j z_j) \sum_{(i,j)\in A} f_{ij} = \sum_{(j,q)\in A} f_{jq} && \forall j \in V' && (8)\\
& 0 \leq f_{ij} \leq u_{ij} && \forall (i,j) \in A && (9)\\
& z_j \in \{0,1\} && \forall j \in V' && (10)
\end{aligned}
$$

As before, the objective function (5) models the number of evacuees entering the safety node, while Constraint (6) bounds the number of evacuees leaving the start node. We use new binary variables z_i to model if node $i \in V'$ is improved or not. Thus, Constraint (7) is a knapsack-constraint on the spendable budget. Finally, Constraints (8) model the loss of flow along the nodes, where the loss factor is $p_i + \hat{p}_i$, if i is improved, and p_i otherwise.

We linearize the products $z_j f_{ij}$ to gain a mixed-integer linear program. To this end, we introduce additional variables $h_j := \sum_{(i,j)\in A} z_j f_{ij}$. The resulting model is then the following:

$$
\begin{aligned}
\text{(FLI)} \quad \max \quad & \sum_{(i,t)\in A} f_{it} && && (11)\\
& \sum_{(s,i)\in A} f_{si} \leq S && && (12)\\
& \sum_{j\in V'} c_j z_j \leq B && && (13)\\
& h_j \leq M z_j && \forall j \in V' && (14)\\
& h_j \leq \sum_{(i,j)\in A} f_{ij} && \forall j \in V' && (15)\\
& p_j \sum_{(i,j)\in A} f_{ij} + \hat{p}_j h_j = \sum_{(j,q)\in A} f_{jq} && \forall j \in V' && (16)\\
& 0 \leq f_{ij} \leq u_{ij} && \forall (i,j) \in A && (17)\\
& z_j \in \{0,1\} && \forall j \in V' && (18)\\
& h_j \geq 0 && \forall j \in V' && (19)
\end{aligned}
$$

The new Constraints (14) and (15) model that $h_j = \sum_{(i,j)\in A} f_{ij}$ if and only if $z_j = 1$. To this end, M must be a large enough constant (when solving this problem, one would choose different constants $M_j = \min\{S, \sum_{(i,j)\in A} u_{ij}\}$ per constraint).

Note that due to Constraint (13), one can easily show weak NP-completeness of FLI by a reduction from the one-dimensional knapsack problem, where we use one node for each item. Each node has $p_i = 0$ and $\hat{p}_i$ equal to the profit of that item. Each node has also a size c_i and the overall size of the knapsack is equal to B. However, we can even show strong NP-completeness via the 3-partition problem. A proof is given in Appendix A.

Theorem 1. *The decision problem of FLI is strongly NP-complete.*

3 Robust Flows

We now extend the FLI problem to include uncertainty in the loss values p. The motivation is to better model that these values can only be estimates, and will never reflect the actual risk on a vertex. Instead, we assume to know only an uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{|V|}$ that contains all possible realizations of p.

We have to decide where to put our improvement resources before we know the realization of p. Then, the actual scenario becomes revealed and the evacuees take the best possible route with respect to our improvements, and the scenario. Following a worst-case approach, the question is: Where should we put the improvements, so that the evacuee flow is maximal in the worst case that may happen?

This approach follows the idea of adjustable robustness (see [BTGGN04]), by assuming that variables z are *here-and-now* (i.e., need to be decided in advance), and variables f, h are *wait-and-see* (i.e., can be decided later, when the scenario is known). A similar two-stage approach can also be found in [LLMS09], where the authors allow the first-stage solution to change once the scenario is known.

3.1 Finite Uncertainty Sets

We first consider the case of a finite set of scenarios $\mathcal{U}^f = \{p^1, \ldots, p^N\}$, and write $\mathcal{N} := \{1, \ldots, N\}$. We assume that $p_i^k + \hat{p}_i \leq 1$ for all $i \in V$, $k \in \mathcal{N}$. The resulting robust two-stage problem can be modeled in the following way:

$$\text{(RFLI)} \qquad \max \min_{k\in\mathcal{N}} \sum_{(i,t)\in A} f_{it}^k \tag{20}$$

$$\sum_{(s,i)\in A} f_{si}^k \leq S \qquad \forall k \in \mathcal{N} \tag{21}$$

$$\sum_{j\in V'} c_j z_j \leq B \tag{22}$$

$$h_j^k \leq M z_j \qquad \forall j \in V', k \in \mathcal{N} \tag{23}$$

$$h_j^k \leq \sum_{(i,j)\in A} f_{ij}^k \qquad \forall j \in V', k \in \mathcal{N} \tag{24}$$

$$p_j^k \sum_{(i,j)\in A} f_{ij}^k + \hat{p}_j h_j^k = \sum_{(j,q)\in A} f_{jq}^k \qquad \forall j \in V', k \in \mathcal{N} \tag{25}$$

$$0 \leq f_{ij}^k \leq u_{ij} \qquad \forall (i,j) \in A, k \in \mathcal{N} \tag{26}$$

$$z_j \in \{0,1\} \qquad \forall j \in V' \tag{27}$$

$$h_j^k \geq 0 \qquad \forall j \in V', k \in \mathcal{N} \tag{28}$$

We shall also write RFLI($\mathcal{U}^f$) when the uncertainty set underlying this formulation needs to be specified.

For every scenario $k \in \mathcal{N}$, we introduce a new set of flows f^k, h^k, which depend on the risk of the respective scenario. The minimum in the objective function can be removed by writing

$$\begin{aligned} &\max\ \alpha \\ &\quad \alpha \leq \sum_{(i,t) \in A} f_{it}^k && \forall k \in \mathcal{N} \\ &\quad (21\text{–}28) \end{aligned}$$

instead. Note that we may consider only the variables z (i.e., the improvements) as decision variables; then, to evaluate the objective of z, one needs to solve a nominal FLI problem for every scenario to determine the resulting worst-case.

3.2 Bounded Uncertainty Sets

3.2.1 Model

We now extend the previous model by considering more complex uncertainty sets. For every loss value p_i, two possible outcomes $\{\underline{p}_i, \overline{p}_i\}$ are given. However, as it would be too pessimistic to assume that the worst-case $\underline{p}_i$ happens at all nodes simultaneously, we follow the spirit of [BS04] and bound the number of vertices that may deviate from their best-case $\overline{p}_i$. We model this as the following uncertainty set:

$$\mathcal{U} = \mathcal{U}(K) = \Big\{ p \in \mathbb{R}^{|V|} : p_i \in \{\underline{p}_i, \overline{p}_i\}\ \forall i \in V,\ |\{i \in V : p_i < \overline{p}_i\}| \leq K \Big\}$$

As before, we assume $\overline{p}_i + \hat{p}_i \leq 1$. Here, $K \in \mathbb{N}$ is a parameter that can be used to control the degree of conservatism of a solution. For $K = |V|$, the uncertainty set $\mathcal{U}$ becomes a Cartesian product, resulting in a large possible set of outcomes. A resulting robust solution will have a relatively small nominal objective value, but a high degree of robustness. For $K = 0$ on the other hand, the uncertainty set $\mathcal{U}$ becomes a best-case singleton, meaning that uncertainty is ignored and a nominal FLI problem needs to be solved. It is in the hands of the practitioner to find a value for K that suits the application, and the outcomes of multiple alternatives may be compared.

This uncertainty set is similar to the one proposed in [BS04], but only uses its extreme points for better tractability. Other robust optimization problems have used similar sets, see, e.g., [BKK11].

Note that one could write RFLI($\mathcal{U}$) directly as a mixed-integer program like (20–28), as $\mathcal{U}(K)$ is also a finite uncertainty set with exponential size. However, this would ignore the structure of the uncertainty set. In the following, we consider more efficient approaches that do not require the inclusion of all scenarios upfront.

3.2.2 Evaluating a solution

We first consider the following subproblem: Given a fixed choice of improvements z, which scenario $p \in \mathcal{U}$ is responsible for the worst output flow? We may model this question as an optimization problem over the losses p.

$$\begin{aligned} WC(z) := \min\ & f^*(p) \\ & p_i = \overline{p}_i + (\underline{p}_i - \overline{p}_i)w_i + \hat{p}_i z_i && \forall i \in V' \\ & \sum_{i \in V'} w_i \leq K \\ & w_i \in \{0,1\} && \forall i \in V' \\ & p_i \geq 0 && \forall i \in V' \end{aligned}$$

We decide for each node i if it is "bad" with the help of the binary variable w_i. We would like to make a choice in a way that $f^*(p)$ is as small as possible, where $f^*(p)$ denotes the optimal outflow under scenario p. This is itself simply a flow with losses problem.

$$\begin{aligned} f^*(p) := \max\ & \sum_{(i,t) \in A} f_{it} && && (29) \\ & \sum_{(s,i) \in A} f_{si} \leq S && && (30) \\ & p_j \sum_{(i,j) \in A} f_{ij} \geq \sum_{(j,q) \in A} f_{jq} && \forall j \in V' && (31) \\ & 0 \leq f_{ij} \leq u_{ij} && \forall (i,j) \in A && (32) \end{aligned}$$

Note that using "$\geq$" in Constraint (31) instead of "=“ does not change the optimal solution of the flow with losses problem. To integrate the computation of $f^*(p)$ into $WC(z)$, we dualize the problem and get:

$$\begin{aligned} \min\ & \sum_{(i,j) \in A} u_{ij}\alpha_{ij} + S\beta \\ & \alpha_{ij} + \gamma_i - p_j\gamma_j + \chi_s(i)\beta \geq \chi_t(j) && \forall (i,j) \in A \\ & \alpha_{ij} \geq 0 && \forall (i,j) \in A \\ & \beta \geq 0 \\ & \gamma_i \geq 0 && \forall i \in V' \\ & \gamma_s = \gamma_t = 0 \end{aligned}$$

where

$$\chi_i(j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{else} \end{cases}$$

We now use this dual formulation to rewrite $WC(z)$.

$$\begin{aligned} \min\ & \sum_{(i,j) \in A} u_{ij}\alpha_{ij} + S\beta \\ & \sum_{i \in V'} w_i \leq K \end{aligned}$$

$$
\begin{aligned}
&\alpha_{ij} + \gamma_i - (\overline{p}_j + (\underline{p}_j - \overline{p}_j)w_j + \hat{p}_j z_j)\gamma_j + \chi_s(i)\beta \geq \chi_t(j) && \forall (i,j) \in A \\
&\alpha_{ij} \geq 0 && \forall (i,j) \in A \\
&\beta \geq 0 \\
&\gamma_i \geq 0 && \forall i \in V' \\
&\gamma_s = \gamma_t = 0 \\
&w_i \in \{0,1\} && \forall i \in V'
\end{aligned}
$$

Due to the product $w_j \gamma_j$, this is a non-linear program (note that z is fixed, and the product $z_j \gamma_j$ is linear here). Substituting $\gamma'_i = w_i \gamma_i$ yields

$$
\begin{aligned}
\min \; & \sum_{(i,j)\in A} u_{ij}\alpha_{ij} + S\beta && && (33)\\
& \sum_{i \in V} w_i \leq K && && (34)\\
& \alpha_{ij} + \gamma_i - (\overline{p}_j + \hat{p}_j z_j)\gamma_j + (\overline{p}_j - \underline{p}_j)\gamma'_j + \chi_s(i)\beta \geq \chi_t(j) && \forall (i,j) \in A && (35)\\
& \gamma'_i \leq \gamma_i && \forall i \in V' && (36)\\
& \gamma'_i \leq M w_i && \forall i \in V' && (37)\\
& \alpha_{ij} \geq 0 && \forall (i,j) \in A && (38)\\
& \beta \geq 0 && && (39)\\
& \gamma_i, \gamma'_i \geq 0 && \forall i \in V' && (40)\\
& \gamma_s = \gamma_t = 0 && && (41)\\
& w_i \in \{0,1\} && \forall i \in V' && (42)
\end{aligned}
$$

where M is a constant large enough ($M \geq |V|$ suffices).

Note that this solves the separation problem of a two-stage problem without fixed recourse (see, e.g., [AP15], for recent general results on this topic).

4 Solution Algorithms

4.1 Scenario Generation

Using formulation (33–42) for $WC(z)$, we can evaluate the objective value for a choice of improvements z and also produce a scenario w where this objective value is attained. In the following, this is used as part of a solution algorithm.

We start with any finite scenario set $\mathcal{U}^0 \subseteq \mathcal{U}$, e.g., $\mathcal{U}^0 = \emptyset$ or $\mathcal{U}^0 = \{\overline{p}\}$. Solving RFLI($\mathcal{U}^0$) yields some solution for the improvements z^0 and an objective value OBJ^0. Solving $WC(z^0)$ determines a new scenario given by w^0 and an objective value WC^0. Setting $\mathcal{U}^1 := \mathcal{U}^0 \cup \{w^0\}$ gives a new scenario set that can be used to determine the next choice of improvements, which is then iteratively repeated.

The algorithm stops with an improvement z^k, when the objective value of the worst-case problem $WC(z^k)$ equals the objective value of RFLI($\mathcal{U}^k$). As each problem RFLI($\mathcal{U}^k$) for a scenario set $\mathcal{U}^k \subseteq \mathcal{U}$ is a relaxation of the problem RFLI($\mathcal{U}$), the optimal objective value of each solution z^k is an upper bound on the optimal objective value of problem RFLI($\mathcal{U}$). Therefore, optimality is proved when the worst-case objective equals the estimated upper bound.

Algorithm 1 (Exact Algorithm for RFLI($\mathcal{U}$))

```
Require: An instance of RFLI($\mathcal{U}$).
 1: $k \leftarrow 0$
 2: $\mathcal{U}^0 \leftarrow \{\overline{p}\}$
 3: Solve RFLI($\mathcal{U}^k$). Let $z^k$ be the resulting solution for the improvements, and
    $OBJ^k$ the resulting objective value.
 4: Solve $WC(z^k)$. Let $p^k$ be the resulting scenario, and $WC^k$ the resulting
    objective value.
 5: if $OBJ^k = WC^k$ then
 6:     return Optimal improvements $z^k$ and objective value $OBJ^k$.
 7: else
 8:     $\mathcal{U}^{k+1} \leftarrow \mathcal{U}^k \cup \{p^k\}$
 9:     $k \leftarrow k + 1$
10:     Goto 3
11: end if
```

The algorithm is summarized as Algorithm 1.

Note that the values OBJ^k are monotonically decreasing, being the optimal objective values to problems that have increasingly more constraints and variables. However, their actual objective values WC^k are not necessarily monotonically increasing. Figure 1 provides an example for a typical run of Algorithm 1. Encircled are solutions which are better than solutions from previous iterations.

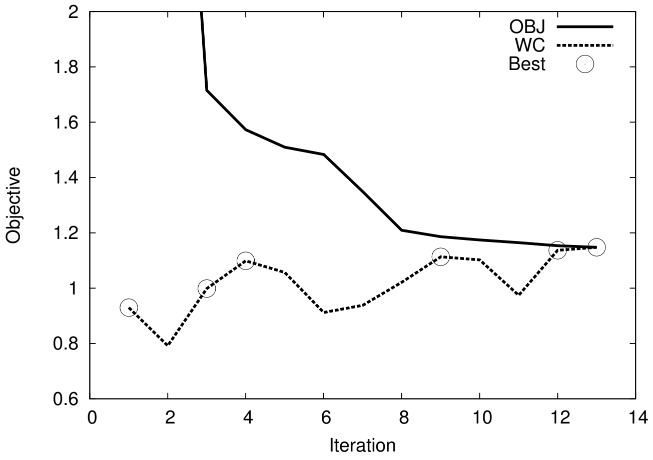


Figure 1: Example run of Algorithm 1. Optimality is reached when OBJ^k and WC^k coincide.

4.2 Artificial Scenarios

As the nominal problem FLI is already NP-hard and the robust version further increases its computational difficulty, we now introduce heuristic solution approaches.

Here we aim at the solution of one FLI problem with a single scenario $p \in \mathbb{R}^{|V|}$ that "represents" the whole uncertainty set $\mathcal{U}$ appropriately. It is not necessary that p is an actually possible scenario, i.e., $p \notin \mathcal{U}$ is possible. While this problem can be solved comparatively fast, its quality depends on how p is determined.

We consider the following cases, setting $n := |V|$.

1. We set $p = \lambda \underline{p} + (1 - \lambda)\overline{p}$ for some constant $\lambda \in [0, 1]$. This includes the best-case scenario $p = \overline{p}$, and the worst-case scenario $p = \underline{p}$.

2. As a special case of this convex combination, we use a value for λ that depends on K and n. If one expects "bad" nodes to be randomly uniformly distributed, then the expected loss is given as $p = \lambda \underline{p} + (1 - \lambda)\overline{p}$ with $\lambda = K/n$.

3. Finally, assumption that "bad" nodes are uniformly distributed may be too optimistic, given that only nodes are relevant which actually carry some flow. Estimating that in a graph with n nodes, only $\sqrt{n}$ nodes have a flow which is larger than zero, we get the more pessimistic value $\lambda = \min\{1, K/\sqrt{n}\}$. If the graph class is known, such an estimate may be further refined.

Any of the above approaches yield a heuristic set of improvements z. To evaluate these solutions, a problem $WC(z)$ still needs to be solved. For large instances, this evaluation may take more computation time than for finding the improvements.

4.3 Rollout and Local Search

As the artificial scenarios approach from Section 4.2 is a one-shot heuristic (i.e., a single solution is constructed), we discuss two ways to use such a solution as a starting point to two different, iterative heuristic methods. Note that to evaluate a choice of improvements z, the mixed-integer program $WC(z)$ needs to be solved. This creates computational difficulties for heuristic procedures which rely on the evaluation of many solutions.

The first approach we describe is a rollout heuristic (see [BTW97] for a general discussion of this method). The idea of this method is to use the method from Section 4.2 as a look-ahead to fix variables heuristically. Rollout heuristics can be seen as a simplification of branch-and-bound algorithms, where only one branch of the tree is considered in depth.

Let us assume that the nodes $j = 1, \ldots, n$ are sorted in some order of importance (e.g., sorted in decreasing order of improvement per cost $\hat{p}_j/c_j$). We begin with heuristically solving two subproblems: We fix $z_1 = 1$, and we fix $z_1 = 0$, i.e., we enforce the improvement of the first node, and we enforce that the first node must not be improved. Both subproblems are solved using the artificial scenario approach.

We then check which solution yields the better objective value and keep the respective variable fixing constraint on z_1. We then proceed to the second node, and so on: Having considered node j, we proceed to node $j + 1$ and repeat the same process.

As there is a budget constraint on the number of improvements, we first check if the remaining improvement budget allows both possibilities, whenever we fix variables in this way. The rollout heuristic ends if either the complete list of nodes is exhausted (which means that linearly many subproblems were solved using the artificial scenario approach), or a timelimit is reached. The best solution we encountered is returned.

Note that even after the first iteration, this approach is at least as good as using the one-shot heuristic (the reason being that the one-shot heuristic does or does not improve the first node). The rollout heuristic uses the one-shot heuristic to explore a larger set of candidate solution using variable branching.

As a second solution approach, we consider a local search on the improvements z. Using the artificial scenario approach to generate a feasible starting solution, we compile a list of feasible neighborhood moves, where a move is to either remove the improvement of a node and add the improvement of a different node, or to add a new improvement without removing an already existing improvement (in both cases, given that the budget suffices). The number of moves is thus bounded by $\mathcal{O}(n^2)$.

We consider the feasible neighborhood moves in random order. To evaluate any move, a MIP needs to be solved, as described in Section 3.2.2. If the evaluation of a move results in an improvement in the objective value, the move is accepted, the current solution updated, and a new list of feasible neighborhood moves is generated. This is repeated until either no improving move is found within the list (i.e., a local optimum is reached), or the timelimit is reached.

Note that this algorithm could also be extended by meta-heuristics such as simulated annealing or tabu search. However, as evaluating a move is computationally expensive, such approaches do not seem promising based on our experimental experience.

As noted above, both the rollout heuristic and the local search need to evaluate several candidate solutions by solving a MIP, which may take too long on large-sized instances for our heuristics to be efficient. Therefore, we consider two additional, heuristic ways to evaluate a solution. The first heuristic solution evaluation simply uses a fixed timelimit on the solution of (33–42) to ensure that a sufficient number of solutions can be evaluated. Our second heuristic evaluation uses a local search on the w variables to determine the objective value of a candidate z solution. Note that this local search is different to the local search on the z variables described above. As a neighborhood move, we chose two variables $w_i = 0$ and $w_j = 1$, and set $w_i = 1$ and $w_j = 0$. Each move is then evaluated by solving a flow with losses problem of type (29–32). Because this evaluation is comparatively fast, we first sample a set of random candidate scenarios to warmstart the search.

4.4 Affine Adjustable Counterpart

We now consider a heuristic approach to solve RFLI based on limiting the degree of freedom for the adjustable variables f and h (as first proposed in [BTGGN04]). If we consider f and h as functions on the scenario set, i.e., $f : \mathcal{U} \to \mathbb{R}^{|A|}$ and $h : \mathcal{U} \to \mathbb{R}^{|V|}$, then problem RFLI does not impose any restrictions on the form these functions may take. For the heuristic affine adjustable counterpart, we demand that both f and h are affine linear functions, i.e.,

$$f_{ij}(p) = \phi_{ij} + \sum_{\ell \in V} \Phi_{ij\ell} p_\ell \qquad h_j(p) = \eta_j + \sum_{\ell \in V} H_{j\ell} p_\ell$$

As we restrict the variables space to affine linear functions, we may not find an optimal solution to RFLI; instead, we solve a conservative approximation.

The reason for using affine linear functions is two-fold. On the one hand, the second-stage variables of many optimization problems show an affine linear behavior (see [BTGGN04]). On the other hand, and in this case more importantly, an advantage of this approach is that the resulting model can be reformulated to a more compact, better tractable form. As an example, we consider the objective function

$$\max \min_{p \in \mathcal{U}} \sum_{(i,t) \in A} f_{it}(p).$$

Substituting f as above, we consider the approximate objective function

$$\max \min_{p \in \mathcal{U}} \sum_{(i,t) \in A} \left(\phi_{it} + \sum_{\ell \in V} \Phi_{it\ell} p_\ell \right)$$

instead. Using the previously used duality technique, we rewrite this objective as

$$\begin{aligned}
\max \quad & \sum_{(i,t) \in A} \left(\phi_{it} + \sum_{\ell \in V} \Phi_{it\ell} \overline{p}_\ell \right) - K\pi^0 - \sum_{\ell \in V} \nu^0_\ell \\
& \sum_{(i,t) \in A} \Phi_{it\ell} (\overline{p}_\ell - \underline{p}_\ell) \leq \pi^0 + \nu^0_\ell && \forall \ell \in V \\
& \pi^0, \nu^0 \geq 0
\end{aligned}$$

with additional variables π^0, ν^0.

Following similar techniques for the constraints of RFLI, the affine adjustable counterpart (AARC) of FLI is given by the following MIP:

$$\max \sum_{(i,t) \in A} \left(\phi_{it} + \sum_{\ell \in V} \Phi_{it\ell} \overline{p}_\ell \right) - K\pi^0 - \sum_{\ell \in V} \nu^0_\ell \tag{43}$$

$$\sum_{(i,t) \in A} \Phi_{it\ell} (\overline{p}_\ell - \underline{p}_\ell) \leq \pi^0 + \nu^0_\ell \qquad \forall \ell \in V \tag{44}$$

$$\sum_{(s,i) \in A} \sum_{\ell \in V} \Phi_{si\ell} \overline{p}_\ell + \sum_{(s,i) \in A} \phi_{si} \leq S \tag{45}$$

$$\sum_{j \in V'} c_j z_j \leq B \tag{46}$$

$$\eta_j + \sum_{\ell \in V} H_{j\ell} \overline{p}_\ell \leq M z_j \qquad \forall j \in V' \tag{47}$$

$$\eta_j + \sum_{\ell \in V} H_{j\ell} \overline{p}_\ell + K\pi^1_j + \sum_{\ell \in V} \nu^1_{j\ell} \leq \sum_{(i,j) \in A} \phi_{ij} + \sum_{\ell \in V} \sum_{(i,j) \in A} \Phi_{ij\ell} \overline{p}_\ell \qquad \forall j \in V \tag{48}$$

$$K\pi^1_j + \nu^1_{j\ell} \geq (\overline{p}_\ell - \underline{p}_\ell) \left(\left(\sum_{(i,j) \in A} \Phi_{ij\ell} \right) - H_{j\ell} \right) \qquad \forall j, \ell \in V \tag{49}$$

$$\underline{p}_j \sum_{(i,j)\in A} \phi_{ij} + \sum_{\ell\in V}\sum_{(i,j)\in A} \Phi_{ij\ell}\underline{p}_j\overline{p}_\ell + \hat{p}_j\eta_j + \sum_{\ell\in V} H_{j\ell}\hat{p}_j\overline{p}_\ell \geq \sum_{(j,q)\in A} \phi_{jq} + \sum_{\ell\in V}\sum_{(j,q)\in A} \Phi_{jq\ell}\overline{p}_\ell + (K-1)\pi_j^2 + \sum_{\ell\in V}\nu_{j\ell}^2 \qquad \forall j \in V \tag{50}$$

$$\pi_j^2 + \nu_{j\ell}^2 \geq (\overline{p}_\ell - \underline{p}_\ell)\left(\left(\sum_{(i,j)\in A}\Phi_{ij\ell}\right)\underline{p}_j - \left(\sum_{(j,q)\in A}\Phi_{jq\ell}\right) + H_{j\ell}\hat{p}_j\right) \qquad \forall j,\ell \in V \tag{51}$$

$$\overline{p}_j \sum_{(i,j)\in A} \phi_{ij} + \sum_{\ell\in V}\sum_{(i,j)\in A} \Phi_{ij\ell}\overline{p}_j\overline{p}_\ell + \hat{p}_j\eta_j + \sum_{\ell\in V} H_{j\ell}\hat{p}_j\overline{p}_\ell \geq \sum_{(j,q)\in A} \phi_{jq} + \sum_{\ell\in V}\sum_{(j,q)\in A} \Phi_{jq\ell}\overline{p}_\ell + K\pi_j^3 + \sum_{\ell\in V}\nu_{j\ell}^3 \qquad \forall j \in V \tag{52}$$

$$\pi_j^3 + \nu_{j\ell}^3 \geq (\overline{p}_\ell - \underline{p}_\ell)\left(\left(\sum_{(i,j)\in A}\Phi_{ij\ell}\right)\overline{p}_j - \left(\sum_{(j,q)\in A}\Phi_{jq\ell}\right) + H_{j\ell}\hat{p}_j\right) \qquad \forall j,\ell \in V \tag{53}$$

$$\phi_{ij} + \sum_{\ell\in V}\Phi_{ij\ell}\overline{p}_\ell \leq u_{ij} \qquad \forall (i,j) \in A \tag{54}$$

$$0 \leq \phi_{ij} \qquad \forall (i,j) \in A \tag{55}$$

$$0 \leq \Phi_{ij\ell} \qquad \forall (i,j) \in A, \ell \in V \tag{56}$$

$$0 \leq \eta_j \qquad \forall j \in V \tag{57}$$

$$0 \leq H_{j\ell} \qquad \forall j,\ell \in V \tag{58}$$

$$z_j \in \{0,1\} \qquad \forall j \in V' \tag{59}$$

$$\pi^i, \nu^i \geq 0 \qquad \forall i = 0,1,2,3 \tag{60}$$

The details on the derivation of AARC are shown in Appendix B.

5 Experiments

We present two sets of experiments. In the first one, we use randomly generated grid graphs to evaluate the algorithms presented in this paper. In the second experiment, we use realistic data based on the city of Nice, France, to compare our heuristic approaches.

All experiments were conducted on a computer with 96GB RAM and a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 20MB cache, and Ubuntu 12.04. Mixed-integer programs were solved using Cplex v. 12.6. and C++ programs compiled with gcc v. 4.5.4. and flag -O3. For the first experiment, one core was used per algorithm. For the second experiment, all 16 cores were used.

5.1 Random Instances

Instances. We created 15 sets $\mathcal{I}_\ell$, $\ell = 4, \ldots, 18$ of random instances to compare our solution algorithms for the RFLI problem. Each graph is a grid of size $\ell \times \ell$, where evacuees start in the lower left corner and need to reach the upper right corner. S is always equal to 100, and B is chosen uniformly randomly from $\{1, \ldots, \ell\}$. We set K to be either 2, 8, or 16 (controlling the degrees of freedom of the uncertainty set). Arc capacities are generated in $\{1, \ldots, 100\}$, and we generate losses $\underline{p}_i \in [0.001, 0.999]$, $\overline{p}_i \in [\underline{p}_i, 1]$, and $\hat{p}_i \in [0, 1 - \overline{p}_i]$ (in this order). Improvement costs c_i are chosen randomly from $[0.5, 1.5]$. For each $\ell = 4, \ldots, 18$, we generated 20 such instances (i.e., a total of $15 \cdot 20 \cdot 3 = 900$ instances).

Setting. Every instance is solved using the iterative Algorithm 1 (which is denoted as EX). Note that Algorithm 1 is not a branch-and-cut algorithm, where scenarios are generated on-the-fly. Instead, problem RFLI is solved to optimality before the worst-case problem WC is used to generate a new scenario.

So we additionally compare this exact solution approach to a lazy constraint generation approach, where we solve a single master RFLI problem, and additional scenarios are generated on-the-fly within the same branch-and-bound tree. We did not implement a branch-and-cut algorithm from scratch, but used the Cplex framework for this purpose. Using Cplex callbacks, it is possible to add cuts during the optimization process. Whenever such a callback routine is called, the worst-case problem WC is solved to optimality to generate the current worst-case scenario. This solution approach is denoted as EX-L.

As two further variants of this lazy constraint method, we analyze the impact of solving the scenario generation problem heuristically using the same evaluation methods described in Section 4.3, i.e., a Cplex callback to generate new cuts will either impose a timelimit on the WC problem, or run the same local search algorithm on w as used for the heuristics in Section 4.3 to find a heuristic cut. When using a timelimit on the evaluation, we refer to the resulting solution as EX-L-TL; when using a local search, we refer to the solution as EX-L-LOC.

Furthermore, we determine heuristic solutions using artificial scenarios as described in Section 4.2 with $\lambda \in \{0.00, 0.50, 1.00, K/n, K/\sqrt{n}\}$. We refer to these solutions as W(λ). We additionally use the rollout and local search methods described in Section 4.3, and refer to these solutions as ROLL-EX (rollout with exact evaluation), ROLL-TL (rollout with a timelimit on the evaluation), ROLL-LOC (rollout with local search for the evaluation), and analogously for LOC-EX, LOC-TL, LOC-LOC. The evaluation timelimit is 10 seconds for instances with $\ell \geq 10$, and 5 seconds for smaller instances. For rollout and local search, we use W($K/\sqrt{n}$) as the underlying heuristic.

Finally, we also solve the affine adjustable robust counterpart, which is denoted as AFF.

We record the objective values of these algorithms using a timelimit of two minutes each. To determine the objective value, we give another two minutes time to solve the evaluation problem.

Results. We summarize our results in Tables 1–4.

The average relative objective value is compared in Tables 1–3. We normalized values using the objective value of the best solution found per instance; e.g., the value 0.73 in the bottom left corner of Table 1 means that that on average, the best objective per instance divided by the objective value of the solution produced with Algorithm 1 equals 73% percent. The best performance in each row is highlighted in bold.

For $K = 2$, the exact, iterative approach from Algorithm 1 shows clearly the best performance of all solution methods. Using lazy constraints performs well for smaller instances, but is not competitive anymore for $\ell \geq 13$ on. A possible reason for this behavior is the computational costliness to solve the separation problem. The heuristics from Table 2 show better performance, with no clear winner over the different variants. The one-shot heuristics from Table 3 show mixed performance, with the more conservative approaches failing to produce good results due to the small value of K. The AARC is computationally difficult to solve and only competitive for small instances.

If we increase K, this overall picture becomes quite different. For larger values of K, Algorithm 1 cannot perform a sufficient number of iterations within the timelimit, and the heuristic approaches becomes more relevant. In particular the rollout heuristic with a fixed timelimit on the evaluation time shows good performance, and also the more conservative one-shot heuristics.

We analyze the performance of EX in more detail in Table 4, where we present the average time spent solving the lower bound problem (33–42), the average number how often this problem is solved, and the same for the corresponding upper bound problems. In the last column, the number of instances is shown where EX succeeded in proving optimality of the produced solution (a value of 20 meaning that all instances of this size could be solved to proven optimality).

These results confirm the strong connection between the size of the uncertainty set, and the efficiency of EX. While for $K = 2$, most of the time is spent in the upper bound problems (which become increasingly hard to solve with every iteration), the lower bound problem consumes most of the computation time for $K = 8$ and $K = 16$.

To summarize these findings, we found that for small values of K, the exact, iterative approach produces high-quality solutions, which are often shown to be optimal. For larger values of K, our heuristic approaches are the better choice; in particular those that bound the computation time spent in solving the lower bound problem.

Regarding computation times of the other algorithms, EX-L and EX-L-TL usually require more time than EX, as is reflected in the resulting objective values. Local search and rollout algorithms use as much computation time as allowed by construction, while the W(λ) algorithms only use the full timelimit for $\mathcal{I}_{18}$, and be generally well below it. AFF requires even more computation time than the exact algorithms for most of the instances.

5.2 Realistic Instances

Instances. We now consider problem instances which are based on real-world data modeling the city of Nice, France. The city is situated in a seismically active region, and has encountered several earthquakes and tsunamis in its history.

K	Instance	EX	EX-L	EX-L-TL	EX-L-LOC
2	$\mathcal{I}_4$	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	**1.00**	0.99	**1.00**	0.99
	$\mathcal{I}_6$	**1.00**	0.99	0.99	**1.00**
	$\mathcal{I}_7$	**1.00**	**1.00**	0.99	0.99
	$\mathcal{I}_8$	**1.00**	0.99	**1.00**	0.98
	$\mathcal{I}_9$	**0.99**	**0.99**	0.98	0.98
	$\mathcal{I}_{10}$	**0.99**	0.97	0.98	0.97
	$\mathcal{I}_{11}$	**0.98**	0.96	0.96	0.96
	$\mathcal{I}_{12}$	0.94	**0.96**	0.95	0.91
	$\mathcal{I}_{13}$	**0.94**	0.78	0.88	0.75
	$\mathcal{I}_{14}$	**0.94**	0.75	0.86	0.78
	$\mathcal{I}_{15}$	**0.98**	0.65	0.80	0.70
	$\mathcal{I}_{16}$	**0.96**	0.51	0.58	0.40
	$\mathcal{I}_{17}$	**0.93**	0.29	0.55	0.33
	$\mathcal{I}_{18}$	**0.95**	0.14	0.20	0.26
	All	**0.97**	0.80	0.85	0.80
8	$\mathcal{I}_4$	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	**0.99**	**0.99**	**0.99**	**0.99**
	$\mathcal{I}_6$	**1.00**	0.99	0.99	0.98
	$\mathcal{I}_7$	**0.98**	0.93	**0.98**	0.96
	$\mathcal{I}_8$	0.94	0.72	0.88	**0.97**
	$\mathcal{I}_9$	**0.93**	0.59	0.80	0.90
	$\mathcal{I}_{10}$	0.87	0.54	0.67	0.82
	$\mathcal{I}_{11}$	0.79	0.46	0.57	0.82
	$\mathcal{I}_{12}$	0.82	0.22	0.45	0.60
	$\mathcal{I}_{13}$	0.81	0.23	0.59	0.53
	$\mathcal{I}_{14}$	0.77	0.27	0.56	0.59
	$\mathcal{I}_{15}$	0.87	0.21	0.61	0.49
	$\mathcal{I}_{16}$	0.79	0.20	0.49	0.41
	$\mathcal{I}_{17}$	0.71	0.12	0.21	0.33
	$\mathcal{I}_{18}$	0.84	0.15	0.27	0.22
	All	0.87	0.51	0.67	0.71
16	$\mathcal{I}_4$	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_6$	0.98	0.97	**0.99**	**0.99**
	$\mathcal{I}_7$	**0.98**	0.88	0.87	0.93
	$\mathcal{I}_8$	0.93	0.61	0.68	0.82
	$\mathcal{I}_9$	0.80	0.47	0.65	0.74
	$\mathcal{I}_{10}$	0.73	0.48	0.41	0.58
	$\mathcal{I}_{11}$	0.67	0.40	0.27	0.52
	$\mathcal{I}_{12}$	0.75	0.18	0.32	0.46
	$\mathcal{I}_{13}$	0.70	0.19	0.38	0.45
	$\mathcal{I}_{14}$	0.71	0.25	0.45	0.59
	$\mathcal{I}_{15}$	0.74	0.18	0.37	0.44
	$\mathcal{I}_{16}$	0.67	0.21	0.39	0.34
	$\mathcal{I}_{17}$	0.58	0.11	0.13	0.25
	$\mathcal{I}_{18}$	0.73	0.16	0.19	0.20
	All	0.80	0.47	0.54	0.62

Table 1: Average normalized objective values for exact solution approaches.

K	Instance	LOC-EX	LOC-TL	LOC-LOC	ROLL-EX	ROLL-TL	ROLL-LOC
2	$\mathcal{I}_4$	0.99	0.99	0.99	0.99	0.99	0.99
	$\mathcal{I}_5$	0.98	0.99	0.98	0.98	0.98	0.97
	$\mathcal{I}_6$	0.98	0.98	0.97	0.96	0.96	0.95
	$\mathcal{I}_7$	0.92	0.95	0.90	0.92	0.92	0.90
	$\mathcal{I}_8$	0.93	0.96	0.87	0.94	0.94	0.88
	$\mathcal{I}_9$	0.86	0.89	0.77	0.90	0.90	0.86
	$\mathcal{I}_{10}$	0.91	0.93	0.82	0.96	0.95	0.91
	$\mathcal{I}_{11}$	0.90	0.92	0.86	0.94	0.95	0.92
	$\mathcal{I}_{12}$	0.83	0.87	0.80	0.87	0.91	0.84
	$\mathcal{I}_{13}$	0.82	0.86	0.80	0.87	0.91	0.83
	$\mathcal{I}_{14}$	0.80	0.83	0.80	0.84	0.87	0.82
	$\mathcal{I}_{15}$	0.80	0.82	0.78	0.82	0.84	0.81
	$\mathcal{I}_{16}$	0.80	0.80	0.78	0.85	0.85	0.80
	$\mathcal{I}_{17}$	0.78	0.80	0.76	0.81	0.82	0.79
	$\mathcal{I}_{18}$	0.82	0.81	0.77	0.81	0.83	0.81
	All	0.88	0.89	0.84	0.90	0.91	0.87
8	$\mathcal{I}_4$	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	**0.99**	**0.99**	0.96	0.98	0.98	0.97
	$\mathcal{I}_6$	0.98	0.98	0.93	0.95	0.95	0.94
	$\mathcal{I}_7$	0.93	0.92	0.86	0.96	0.96	0.91
	$\mathcal{I}_8$	0.85	0.85	0.79	0.90	0.93	0.89
	$\mathcal{I}_9$	0.86	0.87	0.84	0.91	0.92	0.87
	$\mathcal{I}_{10}$	0.81	0.83	0.78	0.76	**0.88**	0.85
	$\mathcal{I}_{11}$	0.84	0.86	0.82	0.82	**0.92**	0.91
	$\mathcal{I}_{12}$	0.79	0.79	0.74	0.71	**0.87**	0.86
	$\mathcal{I}_{13}$	0.82	0.82	0.80	0.72	**0.84**	0.83
	$\mathcal{I}_{14}$	0.88	0.85	0.86	0.88	**0.90**	**0.90**
	$\mathcal{I}_{15}$	0.84	0.81	0.79	0.84	0.87	0.85
	$\mathcal{I}_{16}$	0.81	0.79	0.77	0.81	**0.87**	0.81
	$\mathcal{I}_{17}$	0.84	0.84	0.78	0.84	0.84	0.84
	$\mathcal{I}_{18}$	**0.86**	0.83	0.82	**0.86**	0.81	0.81
	All	0.87	0.87	0.84	0.86	**0.90**	0.88
16	$\mathcal{I}_4$	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_6$	**0.99**	**0.99**	0.98	**0.99**	**0.99**	0.98
	$\mathcal{I}_7$	0.95	0.96	0.90	0.97	**0.98**	0.95
	$\mathcal{I}_8$	0.89	0.92	0.82	0.92	**0.97**	0.94
	$\mathcal{I}_9$	0.91	0.92	0.85	0.96	**0.98**	0.91
	$\mathcal{I}_{10}$	0.89	0.89	0.84	0.80	**0.93**	**0.93**
	$\mathcal{I}_{11}$	0.90	0.91	0.88	0.90	**0.94**	0.92
	$\mathcal{I}_{12}$	0.87	0.88	0.81	0.83	**0.91**	0.90
	$\mathcal{I}_{13}$	0.88	0.85	0.84	0.88	**0.90**	0.89
	$\mathcal{I}_{14}$	0.89	0.82	0.84	0.85	**0.93**	0.86
	$\mathcal{I}_{15}$	0.91	0.86	0.87	0.90	**0.94**	0.92
	$\mathcal{I}_{16}$	0.87	0.84	0.79	0.87	**0.93**	0.89
	$\mathcal{I}_{17}$	0.90	0.84	0.79	0.90	**0.93**	0.91
	$\mathcal{I}_{18}$	0.80	0.78	0.75	0.81	**0.82**	0.80
	All	0.91	0.90	0.86	0.91	**0.94**	0.92

Table 2: Average normalized objective value for rollout and local search.

K	Instance	W(0)	W(0.50)	W(1.00)	W(K/n)	W($K/\sqrt{n}$)	AFF
2	$\mathcal{I}_4$	0.90	0.93	0.91	0.92	0.93	0.90
	$\mathcal{I}_5$	0.90	0.89	0.88	0.89	0.90	0.87
	$\mathcal{I}_6$	0.83	0.85	0.84	0.86	0.88	0.84
	$\mathcal{I}_7$	0.80	0.77	0.78	0.80	0.80	0.78
	$\mathcal{I}_8$	0.85	0.78	0.73	0.86	0.83	0.74
	$\mathcal{I}_9$	0.81	0.79	0.77	0.80	0.76	0.71
	$\mathcal{I}_{10}$	0.80	0.73	0.69	0.79	0.78	0.58
	$\mathcal{I}_{11}$	0.84	0.74	0.71	0.84	0.84	0.47
	$\mathcal{I}_{12}$	0.82	0.69	0.65	0.82	0.78	0.43
	$\mathcal{I}_{13}$	0.84	0.64	0.59	0.83	0.80	0.35
	$\mathcal{I}_{14}$	0.82	0.73	0.64	0.82	0.79	0.27
	$\mathcal{I}_{15}$	0.85	0.70	0.62	0.85	0.80	0.18
	$\mathcal{I}_{16}$	0.79	0.67	0.59	0.80	0.78	0.17
	$\mathcal{I}_{17}$	0.79	0.73	0.62	0.78	0.77	0.11
	$\mathcal{I}_{18}$	0.83	0.65	0.54	0.83	0.81	0.13
	All	0.83	0.75	0.70	0.83	0.82	0.50
8	$\mathcal{I}_4$	0.90	0.94	0.98	0.94	0.98	0.98
	$\mathcal{I}_5$	0.83	0.88	0.92	0.88	0.92	0.91
	$\mathcal{I}_6$	0.76	0.87	0.90	0.80	0.90	0.89
	$\mathcal{I}_7$	0.68	0.76	0.84	0.70	0.84	0.82
	$\mathcal{I}_8$	0.69	0.75	0.79	0.71	0.79	0.77
	$\mathcal{I}_9$	0.67	0.77	0.82	0.67	0.82	0.77
	$\mathcal{I}_{10}$	0.71	0.77	0.80	0.69	0.80	0.64
	$\mathcal{I}_{11}$	0.74	0.81	0.82	0.77	0.84	0.66
	$\mathcal{I}_{12}$	0.83	0.78	0.82	0.84	0.79	0.58
	$\mathcal{I}_{13}$	0.80	0.78	0.80	0.78	0.81	0.42
	$\mathcal{I}_{14}$	0.75	0.86	0.80	0.72	0.88	0.27
	$\mathcal{I}_{15}$	0.87	0.83	0.86	**0.88**	0.84	0.21
	$\mathcal{I}_{16}$	0.78	0.81	0.82	0.77	0.81	0.20
	$\mathcal{I}_{17}$	0.71	0.84	**0.87**	0.75	0.83	0.12
	$\mathcal{I}_{18}$	0.84	0.84	0.80	0.82	**0.86**	0.15
	All	0.77	0.82	0.84	0.78	0.85	0.56
16	$\mathcal{I}_4$	0.91	0.95	**1.00**	**1.00**	**1.00**	**1.00**
	$\mathcal{I}_5$	0.85	0.93	**1.00**	0.96	**1.00**	0.99
	$\mathcal{I}_6$	0.74	0.93	0.98	0.93	0.98	0.98
	$\mathcal{I}_7$	0.63	0.82	0.91	0.76	0.91	0.91
	$\mathcal{I}_8$	0.64	0.80	0.88	0.73	0.88	0.86
	$\mathcal{I}_9$	0.57	0.79	0.90	0.65	0.90	0.84
	$\mathcal{I}_{10}$	0.64	0.78	0.88	0.65	0.88	0.77
	$\mathcal{I}_{11}$	0.65	0.81	0.90	0.70	0.90	0.75
	$\mathcal{I}_{12}$	0.74	0.76	0.87	0.75	0.87	0.66
	$\mathcal{I}_{13}$	0.69	0.80	0.88	0.73	0.88	0.49
	$\mathcal{I}_{14}$	0.69	0.87	0.89	0.70	0.89	0.22
	$\mathcal{I}_{15}$	0.75	0.82	0.90	0.76	0.91	0.18
	$\mathcal{I}_{16}$	0.68	0.80	0.86	0.69	0.86	0.21
	$\mathcal{I}_{17}$	0.57	0.79	0.91	0.60	0.90	0.10
	$\mathcal{I}_{18}$	0.72	**0.82**	**0.82**	0.73	0.80	0.16
	All	0.70	0.83	0.91	0.76	0.90	0.61

Table 3: Average normalized objective values for one-shot heuristics and the AARC.

K	Instance	Time LB	Number LB	Time UB	Number UB	Optimal
2	$\mathcal{I}_4$	0.11	1.9	0.04	2.4	20
	$\mathcal{I}_5$	0.27	2.5	0.15	3.0	20
	$\mathcal{I}_6$	0.64	3.5	1.01	3.8	20
	$\mathcal{I}_7$	1.45	4.0	1.95	4.3	20
	$\mathcal{I}_8$	1.76	3.8	4.15	4.3	20
	$\mathcal{I}_9$	2.76	4.0	20.71	4.3	19
	$\mathcal{I}_{10}$	4.86	4.4	35.50	5.0	15
	$\mathcal{I}_{11}$	6.44	4.2	56.48	4.9	12
	$\mathcal{I}_{12}$	8.59	4.1	77.14	4.7	8
	$\mathcal{I}_{13}$	11.87	3.8	87.53	4.6	5
	$\mathcal{I}_{14}$	13.56	3.0	93.06	3.9	3
	$\mathcal{I}_{15}$	19.26	3.1	83.18	3.8	6
	$\mathcal{I}_{16}$	14.11	2.5	100.75	3.4	1
	$\mathcal{I}_{17}$	16.36	2.0	103.63	3.0	0
	$\mathcal{I}_{18}$	33.23	1.9	86.78	2.8	0
8	$\mathcal{I}_4$	0.06	2.1	0.04	2.7	20
	$\mathcal{I}_5$	0.41	3.6	0.39	4.2	20
	$\mathcal{I}_6$	3.65	6.2	5.61	6.8	20
	$\mathcal{I}_7$	23.53	7.4	20.12	8.0	17
	$\mathcal{I}_8$	58.03	6.3	34.33	6.8	9
	$\mathcal{I}_9$	80.27	4.0	18.67	4.5	5
	$\mathcal{I}_{10}$	111.34	2.8	6.14	2.9	1
	$\mathcal{I}_{11}$	117.23	1.4	2.79	1.4	0
	$\mathcal{I}_{12}$	116.38	1.3	3.63	1.4	0
	$\mathcal{I}_{13}$	117.84	1.1	2.18	1.1	0
	$\mathcal{I}_{14}$	115.56	1.1	4.46	1.1	0
	$\mathcal{I}_{15}$	116.39	1.0	3.62	1.0	0
	$\mathcal{I}_{16}$	113.63	1.1	6.38	1.1	0
	$\mathcal{I}_{17}$	112.79	1.0	7.23	1.0	0
	$\mathcal{I}_{18}$	105.37	1.0	14.64	1.0	0
16	$\mathcal{I}_4$	< 0.01	1.7	0.02	2.0	20
	$\mathcal{I}_5$	0.10	2.9	0.17	3.4	20
	$\mathcal{I}_6$	3.23	6.0	7.93	6.8	19
	$\mathcal{I}_7$	22.63	8.9	35.75	9.6	16
	$\mathcal{I}_8$	57.48	6.3	46.72	6.7	5
	$\mathcal{I}_9$	93.79	4.1	17.90	4.4	3
	$\mathcal{I}_{10}$	118.50	1.5	1.55	1.5	0
	$\mathcal{I}_{11}$	119.28	1.2	0.77	1.2	0
	$\mathcal{I}_{12}$	118.94	1.1	1.10	1.1	0
	$\mathcal{I}_{13}$	117.98	1.0	2.06	1.0	0
	$\mathcal{I}_{14}$	117.06	1.0	2.97	1.0	0
	$\mathcal{I}_{15}$	116.41	1.0	3.61	1.0	0
	$\mathcal{I}_{16}$	114.23	1.0	5.79	1.0	2
	$\mathcal{I}_{17}$	112.74	1.0	7.28	1.0	0
	$\mathcal{I}_{18}$	105.97	1.0	14.04	1.0	0

Table 4: Detailed results for EX.

Figure 2 shows the underlying graph, which is generated from OpenStreetMap data. We used 500 nodes and 1124 edges.

To generate loss values, we used seismic simulation data from [L$^+$14] that simulates an earthquake with similar characteristics as the Ligure earthquake of 1887 with a magnitude of approximately $6.9 M_W$. Based on these simulation results, loss values were estimated corresponding to the expected damage to infrastructure. Figure 3 shows the level of damages considering the Ligure earthquake scenario.

The population data come from the French National Institute for Statistics and Economic Studies *(INSEE)* who estimate the population of each city including Nice and the repartition of the population over the area. Their estimation also takes into account foreigners during holidays.

In order to know the population impacted by the earthquake scenario, the population repartition given for the city of Nice by *INSEE* in 2011 is mapped with the Ligure damages scenario, i.e., Figure 3 in order to know the number of persons that have to be evacuated. We consider a set of 17 starting points from where evacuees get information about their affectation shelters and the safest way to reach them.

The shelter locations are outside the endangered area (i.e., the area in red). In this case study, we consider 7 of them near the living places that have to be evacuated and each of them has a capacity.

No data regarding the costs to improve the safety of locations was available. We therefore generated random values from $[0.5, 1.5]$ per node corresponding to "unit improvement costs", e.g., 10,000 Euros. Such improvements correspond to seismic retrofitting measures, such as the construction of stabilizing trusses on endangered buildings.

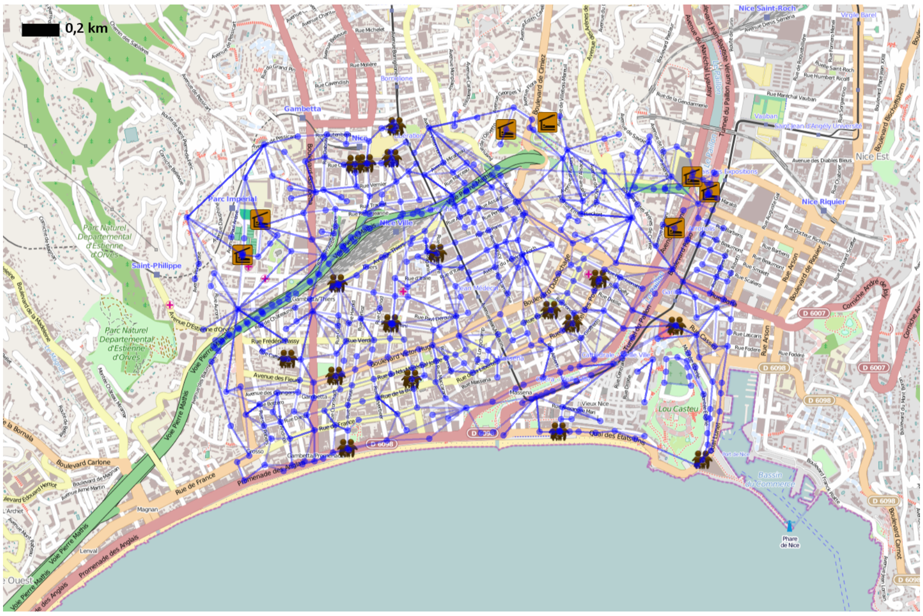


Figure 2: Graph for Nice.

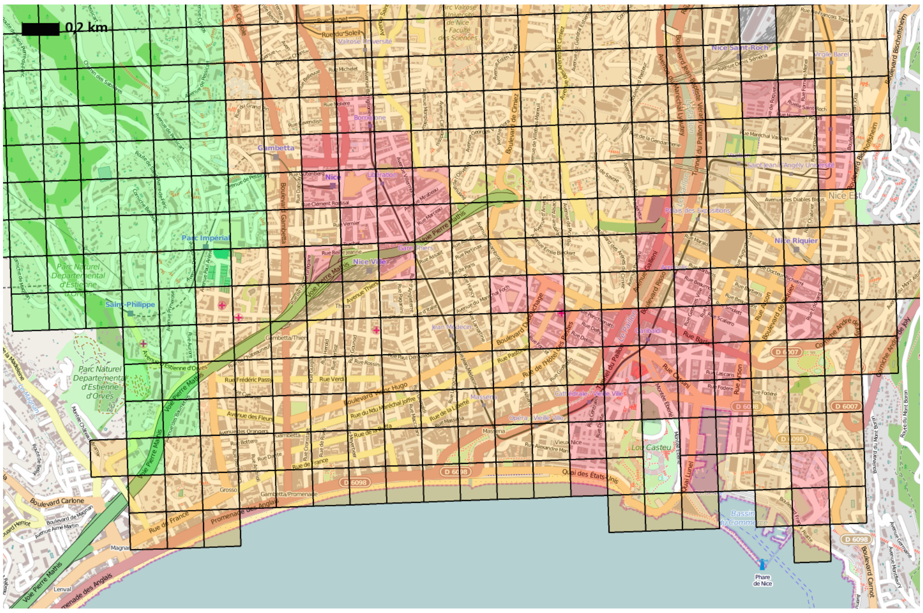


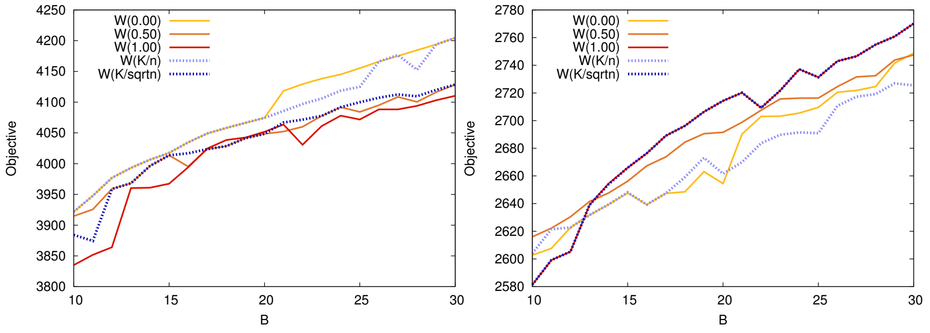
Figure 3: Ligure damages scenario for Nice

Setting. We consider the two scenario sets $\mathcal{U}(10)$ (i.e., only 2% of all nodes may have their worst-case loss value) and the more conservative set $\mathcal{U}(50)$ (i.e., 10% of nodes). We solved these instances heuristically using $\lambda \in \{0.00, 0.50, 1.00, K/n, K/\sqrt{n}\}$ for varying improvement budget $B \in [10, 30]$. To limit computation times, we imposed a timelimit of 120 seconds for each lower bound problem.

Results. We compare the heuristic objective values for $K = 10$ and $K = 50$ in Figures 4(a) and 4(b), respectively. While the more optimistic estimates $\lambda = 0.00$ and $\lambda = K/n$ outperform the others for $K = 10$, this is turned around for $K = 50$. Overall, the value $\lambda = K/\sqrt{n}$ provides a good compromise choice for both settings. Computation times for the upper bound problems were in the order of a few seconds, while the lower bound problems always required the full 120 seconds available.

Using computational results as presented in Figure 4, the planner is provided a helpful tool to decide two questions: Firstly, how many evacuees can be estimated to reach shelter locations under different degrees of damage severeness? Using our population data, estimates lie between ~ 2500 and ~ 4200 evacuees. Secondly, what is the safety gain by increasing the spending on security measures? These results indicate between 8 evacuees per improvement unit for the heavy-damage case, and around 15 evacuees per improvement for the light-damage case. These results may serve as indicators to better prepare for an emergency situation.

Using the affine adjustable heuristic on these instances, no other than the trivial zero-solution could be found within the timelimit.



(a) $K = 10$

(b) $K = 50$

Figure 4: Heuristic objective values for varying improvement budget B.

6 Conclusion

In this paper we contributed to the current literature on network flow evacuation planning models, by introducing a new evacuation model that includes losses along nodes, improvability, and uncertainty. As the scenario set and thus the model is of exponential size, it cannot be solved directly using a mixed-integer programming solver. We therefore developed an algorithm that iteratively increases the problem size by finding the worst-case scenario for the current (partial) solution along with several heuristic methods to construct feasible solutions and to evaluate them.

In computational experiments we compared the performance of these solution approaches using both randomly generated and realistic instances based on the city of Nice, France. Our results show the superior performance of heuristic approaches for instances with large number of nodes and large uncertainty sets. Using the models and methods presented in this paper, a planner is given a tool to estimate the impact of a catastrophe under different spending budgets.

Acknowledgement

We thank the anonymous referees for their helpful comments that helped shaping this paper.

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.

[ACF$^+$13] A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856 – 866, 2013.

[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.

[AP15] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope, 2015. Available online under `http://www.optimization-online.org/DB_HTML/2015/11/5207.html`.

[BCP14] A. Billionnet, M.-C. Costa, and P.-L. Poirion. 2-stage robust MILP with continuous recourse variables. *Discrete Applied Mathematics*, 170(0):21 – 32, 2014.

[BKK11] C. Büsing, A. M. C. A. Koster, and M. Kutschka. Recoverable robust knapsacks: γ-scenarios. In *Network Optimization*, pages 583–588. Springer, 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.

[BTGGN04] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

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

[BTW97] D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu. Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262, 1997.

[CFS82] L.G. Chalmet, R.L. Francis, and P.B. Saunders. Network models for building evacuation. *Fire Technology*, 18(1):90–113, 1982.

[CHT88] W. Choi, H.W. Hamacher, and S. Tufekci. Modeling of building evacuation problems by network flows with side constraints. *European Journal of Operational Research*, 35(1):98 – 110, 1988.

[CLZ06] A. M. Campbell, T. J. Lowe, and L. Zhang. Upgrading arcs to minimize the maximum travel time in a network. *Networks*, 47(2):72–80, 2006.

[DLG11] B. Dilkina, K. J. Lai, and C. P. Gomes. Upgrading shortest paths in networks. In T. Achterberg and J. C. Beck, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 6697 of *Lecture Notes in Computer Science*, pages 76–91. Springer Berlin Heidelberg, 2011.

[DNW04] I. Demgensky, H. Noltemeier, and H.-C. Wirth. Optimizing cost flows by edge cost and capacity upgrade. *Journal of Discrete Algorithms*, 2(4):407 – 423, 2004. The 26th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2000).

[GLMR14] V. Gabrel, M. Lacroix, C. Murat, and N. Remli. Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164, Part 1(0):100 – 111, 2014.

[GS15] M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. *LNCS State-of-the-Art Surveys Springer*, 2015. To appear.

[HHK$^+$11] H.W. Hamacher, S. Heller, W. Klein, G. Köster, and S. Ruzika. A sandwich approach for evacuation time bounds. In R. D. Peacock, E. D. Kuligowski, and J. D. Averill, editors, *Pedestrian and Evacuation Dynamics*, pages 503–513. Springer US, 2011.

[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.

[KMN$^+$98] S. O. Krumke, M. V. Marathe, H. Noltemeier, R. Ravi, and S.S. Ravi. Network improvement problems. *Network Design: Connectivity and Facilities Location, AMSDIMACS Volume Series in Discrete Mathematics and Theoretical Computer Science*, 40:247–268, 1998.

[KY97] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, 1997.

[L$^+$14] A. Lemoine et al. Pligurian earthquake: Seismic and tsunami scenario modeling, from hazard to risk assessment towards evacuations planning. In *Proceedings of the Second European Conference on Earthquake Engineering and Seismology*, 2014.

[LKN11] G. Lämmel, H. Klüpfel, and K. Nagel. Risk minimizing evacuation strategies under uncertainty. In R. D. Peacock, E. D. Kuligowski, and J. D. Averill, editors, *Pedestrian and Evacuation Dynamics*, pages 287–296. Springer US, 2011.

[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.

[LM13] Y. Lin and K. Mouratidis. Best upgrade plans for large road networks. In M. A. Nascimento, T. Sellis, R. Cheng, J. Sander, Y. Zheng, H.-P. Kriegel, M. Renz, and C. Sengstock, editors, *Advances in Spatial and Temporal Databases*, volume 8098 of *Lecture Notes in Computer Science*, pages 223–240. Springer Berlin Heidelberg, 2013.

[NNL$^+$14] I. A. Ndiaye, E. Neron, A. Linot, N. Monmarche, and M. Goerigk. A new model for macroscopic pedestrian evacuation planning with safety and duration criteria. *Transportation Research Procedia*, 2(0):486 – 494, 2014. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.

[Old01] J. D. Oldham. Combinatorial approximation algorithms for generalized flow problems. *Journal of Algorithms*, 38(1):135 – 169, 2001.

[OMH09] S. Opasanon and E. Miller-Hooks. The safest escape problem. *Journal of the Operational Research Society*, 60:1749–1758, 2009.

[OZ07] F. Ordez and J. Zhao. Robust capacity expansion of network flows. *Networks*, 50(2):136–145, 2007.

[Rad98] T. Radzik. Faster algorithms for the generalized network flow problem. *Mathematics of Operations Research*, 23(1):69–100, 1998.

[SK98] S. Schwarz and S.O. Krumke. On budget-constrained flow improvement. *Information Processing Letters*, 66(6):291 – 297, 1998.

[Way99] K. D. Wayne. *Generalized Maximum Flow Algorithms.* PhD Thesis. Cornell University, New York, United States, 1999.

[XLW10] C. Xie, D.-Y. Lin, and S. T. Waller. A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):295 – 316, 2010.

[XT11] C. Xie and M. A. Turnquist. Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. *Transportation Research Part C: Emerging Technologies*, 19(1):40 – 63, 2011.

[Yam96] T. Yamada. A network flow approach to a city emergency evacuation planning. *International Journal of Systems Science*, 27(10):931–936, 1996.

[ZZ13] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457 – 461, 2013.

A Proof of Theorem 1

Theorem 2. *FLI is strongly NP-complete, even if $c_i = 1$ for all $i \in V'$.*

Proof. Given a decision instance of FLI, it requires a polynomial number of operations to verify each constraint in order to answer *Yes* or *No.* Therefore, FLI is in NP.

To show NP-hardness, we consider the 3-partition problem, which is known as strongly NP-hard: Given a (multi)set M of $3N$ natural numbers a_i with $B/4 < a_i < B/2$ for all $i = 1, \dots, 3N$, where $B := 1/N \sum_{i=1}^{3N} a_i$. Find a partition of M into N sets $M_1, \dots, M_N$ such that the sum of numbers a_i in each set is equal to B.

Given an instance of 3-partition, we create an instance of FLI in the following way:

There is a source node s where NB evacuees start. Additionally, there is one node v_i for each $i = 1, \dots, 3N$, one node v'_k for each $k = 1, \dots, N$, nodes v''_{ik} for each $i = 1, \dots, 3N$ and $k = 1, \dots, N$, and one sink node t.

We have $p_i = 1$ and $\hat{p}_i = 0$ for all nodes v_i and v'_i; and $p_{ik} = 0$, $\hat{p}_{ik} = 1$ for all nodes v''_{ik}.

Furthermore, there are arcs from s to v_i for all $i = 1, \dots, 3N$ with capacity a_i; arcs from v_i to v''_{ik} and from v''_{ik} to v'_k with capacity a_i for all $i = 1, \dots, 3N$ and $k = 1, \dots, N$; and arcs from v'_k to t with capacity B for all $k = 1, \dots, N$. An example is presented in Figure 5.

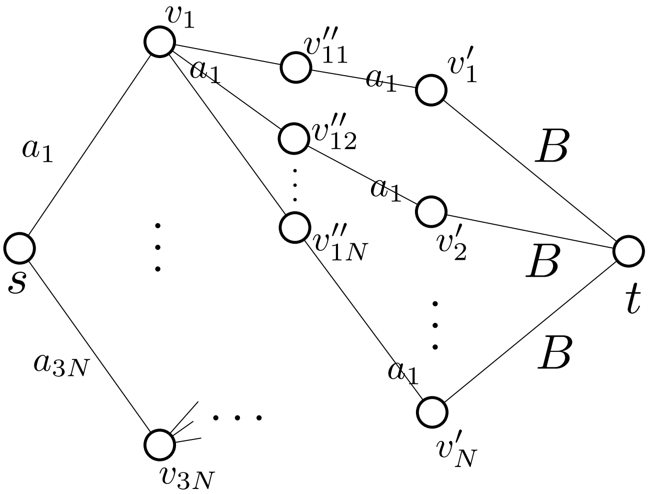


Figure 5: FLI problem built from 3-partition instance.

The improvement costs c_i are equal to 1 for all nodes, and the improvement budget equals N. We have that there exists a 3-partition if and only if the FLI problem has a solution with objective value NB.

To see this, let a 3-partition be given. If value a_i is in set M_k, then we improve node v''_{ik}, meaning the a_i units of flow can go from v_i to v'_k. As we have a 3-partition, there is exactly one possible path between every pair of nodes v_i and v'_k, and the total inflow in the nodes v'_k equals B. Thus, we have a choice of improvements that yields a flow with value NB.

On the other hand, let an optimal solution to the FLI instance with value NB be given. As NB units of flow start from s, there is no loss in the network. This means that for every node v_i, $i = 1, \dots, 3N$, there is exactly one improved node v''_{ik} (as there needs to be at least one such node to prevent loss, and the improvement budget enforces that not more than one such node can be improved). The node improvements can thus be considered as a partitioning of the elements $a_1, \dots, a_{3N}$ into N sets. As each node v'_k has B units of outgoing flow, this partition is a 3-partition with equal sums.

□

B Derivation of AARC

In this section we describe the construction of problem formulation AARC from Section 4.4 in more detail. To this end, we take the RFLI problem with uncertainty set $\mathcal{U}(K)$ and substitute

$$f_{ij}(p) = \phi_{ij} + \sum_{\ell \in V} \Phi_{ij\ell} p_\ell \qquad h_j(p) = \eta_j + \sum_{\ell \in V} H_{j\ell} p_\ell.$$

Using this parameterization of f and h, problem RFLI becomes:

$$\max \min_{p \in \mathcal{U}} \sum_{(i,t) \in A} \left(\phi_{it} + \sum_{\ell \in V} \Phi_{it\ell} p_\ell \right) \tag{61}$$

$$\sum_{(s,i) \in A} \left(\phi_{si} + \sum_{\ell \in V} \Phi_{si\ell} p_\ell \right) \leq S \qquad \forall p \in \mathcal{U} \tag{62}$$

$$\sum_{j \in V'} c_j z_j \leq B \tag{63}$$

$$\eta_j + \sum_{\ell \in V} H_{j\ell} p_\ell \leq M z_j \qquad \forall j \in V', p \in \mathcal{U} \tag{64}$$

$$\eta_j + \sum_{\ell \in V} H_{j\ell} p_\ell \leq \sum_{(i,j) \in A} \left(\phi_{ij} + \sum_{\ell \in V} \Phi_{ij\ell} p_\ell \right) \qquad \forall j \in V', p \in \mathcal{U} \tag{65}$$

$$p_j \sum_{(i,j) \in A} \left(\phi_{ij} + \sum_{\ell \in V} \Phi_{ij\ell} p_\ell \right) + \hat{p}_j \left(\eta_j + \sum_{\ell \in V} H_{j\ell} p_\ell \right)$$

$$\geq \sum_{(j,q) \in A} \left(\phi_{jq} + \sum_{\ell \in V} \Phi_{jq\ell} p_\ell \right) \qquad \forall j \in V', p \in \mathcal{U} \tag{66}$$

$$\phi_{ij} + \sum_{\ell \in V} \Phi_{ij\ell} p_\ell \leq u_{ij} \qquad \forall (i,j) \in A, p \in \mathcal{U} \tag{67}$$

$$0 \leq \phi_{ij} \qquad \forall (i,j) \in A \tag{68}$$

$$0 \leq \Phi_{ij\ell} \qquad \forall (i,j) \in A, \ell \in V \tag{69}$$

$$0 \leq \eta_j \qquad \forall j \in V \tag{70}$$

$$0 \leq H_{j\ell} \qquad \forall j, \ell \in V \tag{71}$$

$$z_j \in \{0,1\} \qquad \forall j \in V' \tag{72}$$

The structure of problem (61-72) is that of a strictly robust problem, i.e., there are no more adjustable variables present. Note that the recourse matrix of RFLI is affected by uncertainty (i.e., adjustable variables are multiplied with uncertain parameters, see Constraint (25) and the affine adjustable version (66). Hence, the rich theory on fixed-recourse problems (see [BTGGN04]) cannot be applied directly.

In the following, we reformulate the constraints that hold for all $p \in \mathcal{U}$ to compact counterparts.

Using the previously used duality technique, we rewrite the objective

$$\max \min_{p \in \mathcal{U}} \sum_{(i,t) \in A} \left(\phi_{it} + \sum_{\ell \in V} \Phi_{it\ell} p_\ell \right)$$

as

$$\begin{aligned}
&\max \sum_{(i,t)\in A} \left(\phi_{it} + \sum_{\ell\in V} \Phi_{it\ell}\overline{p}_\ell\right) - K\pi^0 - \sum_{\ell\in V} \nu^0_\ell \\
&\sum_{(i,t)\in A} \Phi_{it\ell}(\overline{p}_\ell - \underline{p}_\ell) \le \pi^0 + \nu^0_\ell \qquad\qquad \forall \ell \in V \\
&\pi^0, \nu^0 \ge 0
\end{aligned}$$

with additional variables π^0, ν^0.

For Constraints (62), (64) and (67), the worst-case scenario is given by $\overline{p}$, and no additional constraints or variables are required.

The Constraints (65)

$$\eta_j + \sum_{\ell\in V} H_{j\ell} p_\ell \le \sum_{(i,j)\in A} \left(\phi_{ij} + \sum_{\ell\in V} \Phi_{ij\ell} p_\ell\right) \qquad \forall p \in \mathcal{U}$$

have the robust counterpart

$$\begin{aligned}
&\eta_j + \sum_{\ell\in V} \left(H_{j\ell} - \sum_{(i,j)\in A} \Phi_{ij\ell}\right) \overline{p}_\ell + K\pi^1_j + \sum_{\ell\in V} \nu^1_{j\ell} \le \sum_{(i,j)\in A} \phi_{ij} \qquad \forall j \in V \\
&K\pi^1_j + \nu^1_{j\ell} \ge \left(H_{j\ell} - \left(\sum_{(i,j)\in A} \Phi_{ij\ell}\right)\right) (\underline{p}_\ell - \overline{p}_\ell) \qquad \forall j, \ell \in V \\
&\pi^1, \nu^1 \ge 0
\end{aligned}$$

Finally, we consider Constraints (65), which are quadratic in the uncertainty p:

$$\begin{aligned}
&p_j \sum_{(i,j)\in A} \left(\phi_{ij} + \sum_{\ell\in V} \Phi_{ij\ell} p_\ell\right) + \hat{p}_j \left(\eta_j + \sum_{\ell\in V} H_{j\ell} p_\ell\right) \\
&\ge \sum_{(j,q)\in A} \left(\phi_{jq} + \sum_{\ell\in V} \Phi_{jq\ell} p_\ell\right) \qquad \forall j \in V', p \in \mathcal{U}
\end{aligned}$$

As the loss p_j is either $\underline{p}_j$ or $\overline{p}_j$, we use a case distinction to arrive at two sets of robust constraints. For $\underline{p}_j$, we have the compact formulation

$$\begin{aligned}
&\underline{p}_j \sum_{(i,j)\in A} \phi_{ij} + \hat{p}_j \eta_j \\
&\ge \sum_{(j,q)\in A} \phi_{jq} + \sum_{\ell\in V} \left(\left(\sum_{(j,q)\in A} \Phi_{jq\ell}\right) - \left(\sum_{(i,j)\in A} \Phi_{ij\ell}\right) \underline{p}_j - H_{j\ell}\hat{p}_j \right) \overline{p}_\ell \\
&+ (K-1)\pi^2_j + \sum_{\ell\in V} \nu^2_{j\ell} \qquad \forall j \in V \\
&\pi^2_j + \nu^2_{j\ell} \ge \left(\left(\sum_{(j,q)\in A} \Phi_{jq\ell}\right) - \left(\sum_{(i,j)\in A} \Phi_{ij\ell}\right) \underline{p}_j - H_{j\ell}\hat{p}_j \right) (\underline{p}_\ell - \overline{p}_\ell) \qquad \forall j, \ell \in V
\end{aligned}$$

$\pi^2, \nu^2 \geq 0$

while for $p_j = \overline{p}_j$, we use

$$\overline{p}_j \sum_{(i,j)\in A} \phi_{ij} + \hat{p}_j \eta_j$$

$$\geq \sum_{(j,q)\in A} \phi_{jq} + \sum_{\ell \in V} \left(\left(\sum_{(j,q)\in A} \Phi_{jq\ell} \right) - \left(\sum_{(i,j)\in A} \Phi_{ij\ell} \right) \overline{p}_j - H_{j\ell}\hat{p}_j \right) \overline{p}_\ell$$

$$+ K\pi_j^3 + \sum_{\ell \in V} \nu_{j\ell}^3 \qquad \forall j \in V$$

$$\pi_j^3 + \nu_{j\ell}^3 \geq \left(\left(\sum_{(j,q)\in A} \Phi_{jq\ell} \right) - \left(\sum_{(i,j)\in A} \Phi_{ij\ell} \right) \overline{p}_j - H_{j\ell}\hat{p}_j \right) (\underline{p}_\ell - \overline{p}_\ell) \qquad \forall j, \ell \in V$$

$\pi^3, \nu^3 \geq 0$