



A Robust Optimization Approach to Flow Decomposition

Moritz Stinzendörfer^{1,2} · Philine Schiewe¹ · Fabricio Oliveira^{1,3}

Received: 16 October 2025 / Accepted: 4 May 2026 / Published online: 29 May 2026

© The Author(s) 2026

Abstract

In this paper, we generalize the minimum flow decomposition problem (MFD) to incorporate uncertain edge capacities and tackle it from the perspective of robust optimization. In the classical flow decomposition problem, a network flow is decomposed into a set of weighted paths from a fixed source node to a fixed sink node that precisely represents the flow distribution across all edges. MFD problems permeate multiple important applications, including reconstructing genomic sequences to representing the flow of goods or passengers in distribution networks. Inspired by these applications, we generalize the MFD to an inexact case with bounded flow values, provide a detailed analysis, and explore different variants that are solvable in polynomial time. Moreover, we introduce the concept of robust flow decomposition by incorporating uncertain bounds and applying different robustness concepts to handle the uncertainty. Finally, we present two different adjustably robust problem formulations and perform computational experiments illustrating the benefit of adjustability.

Keywords Inexact flow networks · Minimum flow decomposition · Robust optimization · Adjustable robustness · Combinatorial optimization

1 Introduction

Network flows are a fundamental concept in mathematics, particularly in the field of graph theory and optimization. A network is typically represented as a directed graph, with nodes representing junctions or points of interest and edges representing the pathways or connections between these nodes. The primary goal in network flow

✉ Fabricio Oliveira
fabol@dtu.dk

¹ Department of Mathematics and Systems Analysis, Aalto University, Otakaari 1, 02150 Espoo, Finland

² Department of Mathematics, RPTU Kaiserslautern-Landau, Paul-Ehrlich-Str. 14, 67663 Kaiserslautern, Germany

³ DTU Management, Technical University of Denmark, Akademivej 358, 2800 Kongens Lyngby, Denmark

problems is to determine an optimal way to send a certain amount of flow from a source node to a sink node or find a minimum-cost flow. In the latter, each edge has a cost associated with transporting flow through it, and the objective is to minimize the total cost while satisfying flow requirements. These network flow problems have wide-ranging applications in fields such as transportation [9], telecommunications [27], logistics [39], and bioinformatics [51], making them an essential area of study in both theoretical and applied mathematics.

Another important network flow problem is the *minimum flow decomposition* (MFD), which involves decomposing the flow into a set of weighted paths that, once combined, recover how the flow is distributed through the network. This problem structure is present in bioinformatics applications such as multiassembly problems [15], which involve reconstructing genomic sequences from short substrings. For example, in RNA transcript assembly, sequenced substrings from a given sample are used to construct weighted graphs (known as splice or splicing graphs) where nodes represent exons (i.e., the genetic information that remains in the messenger RNA) and edges represent connections between exons. From the splice graph, one seeks to infer how many transcripts (sequences of exons) there are in the splice graph (i.e., weighted paths) such that their superposition recovers (or explains) the given splice graph. However, real-world data often is marred with imprecision and measurement errors, making it challenging to form accurate flow networks, which is why the related research focuses on robustness against measurement errors [14].

Another sector where MFDs are also prevalent is transportation planning. In particular, public transport planning involves several stages, including line planning, timetabling, and vehicle scheduling. The latter stage, in particular, can benefit from the MFD by decomposing trip graphs (i.e., graphs representing the aggregated number of trips between nodes that represent locations) into a minimal number of paths, thereby optimizing vehicle utilization and reducing operational costs. Despite the extensive literature on robustness considerations in public transport planning, most research focuses on timetabling and delay management, considering uncertainties such as travel times or disruptions. On the other hand, demand uncertainties are often addressed in applications focusing on more operational levels, such as vehicle routing problems.

The above applications share as a common feature the latent need of being able to incorporate robustness requirements in their modeling, so that their recommendations are reliable in the face of the uncertainty in their input data. Aiming to address this need, in this paper, we explore the minimum flow decomposition problem (MFDP) from a robust optimization perspective and propose the notion of *robust flow decomposition*. To the best of our knowledge, this is the first work in this direction.

Our main contributions can be summarized as follows:

- We examine existing variants of MFDP with regard to different robustness concepts.
- We generalize the MFDP to the weighted inexact case with lower and upper bounds on the flow values, investigate the complexity of the resulting problem, and explore different variants that are solvable in polynomial time.
- We introduce the concept of robust flow decomposition by incorporating uncertain flows and discuss special cases with different robustness concepts.

- We present two different adjustable problem formulations for which we develop a proof of concept, highlighting the benefit of adjustability in the uncertain case in a subsequent computational study.

The remainder of this paper is structured as follows: The connections to the related literature are discussed in Sect. 2, while in Sect. 3, we present the underlying technical background concerning the MFDP. In Sect. 4, we introduce the generalized deterministic problem considering inexact flows and provide complexity analyses of relevant variants. In Sect. 5, we develop robust minimum flow decomposition problems, starting with strict robustness in Sect. 5.2. Further, we propose two adjustable problem formulations in Sect. 5.3 followed by a corresponding computational study in Sect. 6. Section 7 concludes the paper.

2 Literature Review

The task of optimally covering nodes, edges, or paths in a graph is a common problem that appears in many well-studied concepts in areas of graph theory and network analysis. One example is to find a *minimum path cover (MPC)*, i.e., a set of directed paths with minimum cardinality that covers all vertices (or edges), in a digraph [16]. Since the MPC consists of one path if and only if there is a Hamiltonian path in the corresponding graph, the MPC problem is NP-hard.

However, unlike the MFDP and its robust variants we investigate, the edges only need to be covered by a minimum number of paths, and there are no flow values that have to match weights associated with the paths. Therefore, the MPC can be solved in polynomial time if the graph is acyclic, e.g., by transforming it into a matching problem [33].

The *flow decomposition problem (FDP)* [1], on the other hand, describes the problem of decomposing a network flow into a set of weighted paths from a fixed source node to a fixed sink node such that their union precisely represents the flow distribution across all edges. One way to compute such a flow decomposition in a *directed acyclic graph (DAG)* is to iteratively remove weighted paths that utilize at least one edge. The authors of [1] show that this results in a flow decomposition with at most $|E|$ paths (where $|E|$ is the number of edges in the graph) and can be computed in polynomial time. Nonetheless, when aiming to minimize the number of paths in the decomposition, referred to as *minimum flow decomposition (MFD)*, the problem becomes NP-hard [47] and is even hard to approximate [22]. As a result, efficient heuristics have been developed, such as the greedy methods in [47] or an improved version in [44]. In the former, the proposed algorithms iteratively choose the shortest path or the path with the largest flow value in the remaining flow until the entire flow is decomposed, while in the latter, the flow graph is first modified before applying the greedy method.

Recently, the authors of [15] presented an exact solution approach for the MFDP on DAGs, which is based on an integer linear programming (ILP) approach using only a quadratic number of variables instead of enumerating all possible paths in the network (which would lead to an exponential number of variables). They show

that their proposed method consistently solves instances on both simulated and real datasets significantly faster than previous approaches. These data sets originate from applications in bioinformatics, such as *multiassembly problems*, where MFD plays a key role [52].

2.1 Inexact Flow Decomposition in Transcript Assembly

The multiassembly problem involves reconstructing various genomic sequences from short substrings (called sequenced *reads*) [52], such as RNA transcript assembly [28, 44], i.e., recovering the set of full-length transcripts. Initially, reads are used to construct a weighted graph, namely a splice graph in RNA transcript assembly, where nodes represent exons, and edges denote connections between them. The nodes and edges are weighted, indicating, e.g., their relative abundance in the reads. The multi-assembly problem is then finding a set of weighted paths that best explain the graph's weights [44, 46], and it has been shown that the MFD has a very high accuracy on perfect data [14].

However, the assumption of perfect data does not hold in practice. Due to different sources of uncertainty and measurement errors, splice graphs derived from experimental data are unlikely to form a flow network [51]. To address this problem, the authors of [51] assign intervals of possible weights to the edges instead of exact weights, which forms a so-called *inexact flow network*. The resulting problem of identifying a minimal set of paths that explain these intervals is called the *minimum inexact flow decomposition problem (MIFDP)*, which is also considered in [15]. Other approaches aim to identify a minimum set of weighted paths that minimizes the sum of squared differences between the weight of each edge and the sum of weights of paths passing through it or to assign slack variables to each path to move error handling away from the individual edges [14].

All these approaches have in common that they try to be robust against possible measurement errors. This means that although all data is known, it is likely to be error-prone, which is why, in most cases, no classical flow decomposition can be achieved. However, to ensure that the paths in the decomposition form feasible flows, the requirement that the measured weight of each edge and the sum of weights of paths passing through it must match is relaxed. In contrast to the typical assumptions in robust optimization, a scenario or the real data is not revealed here. While this is suitable in the context of bioinformatics applications, robustness in the classical sense of finding a feasible solution that is robust for every possible realization in an uncertainty set is better suited to enforce robustness requirements in transportation problems or public transport planning, where the MFD also plays a key role.

2.2 Inexact Flow Decomposition in Public Transportation Planning

In public transport planning, the operational costs are mainly determined by the *vehicle schedule* [40], which is part of the traditional sequential planning approach [21, 23]. Typically, this begins on a strategic level with a demand analysis to build passenger

demand matrices, which are then used to define the overall structure of the transit network, including necessary infrastructure such as stations or depots.

In the next step, called *line planning*, the types of services (e.g., bus, rail, tram) and their routes, stops, and frequencies are determined to meet the demand and policy goals [43]. This is followed by the operational planning stage, where the *timetabling* step focuses on creating detailed schedules for each line, resulting in so-called *trips* with departure and arrival times for each station as well as start and end stations [30]. Then, vehicle scheduling allocates vehicles to trips, ensuring that all scheduled trips are covered while optimizing vehicle utilization, often characterized by minimal fleet size or operational costs [9]. The former can be addressed by decomposing the trip graph into a minimal number of paths [9], and also in recent applications such as electric buses with limited driving ranges, decomposition methods are used for vehicle scheduling [34].

Although there is a large body of literature dedicated to robustness in public transport, most of it concentrates on timetabling and delay management by considering uncertain travel times or disruptions [11, 29, 36]. Demand uncertainties are often considered in earlier stages, e.g., stop planning [10], or in other applications such as *vehicle routing problems* [25].

2.3 Robust Optimization

The concept of *robust optimization* was introduced by [4, 45], which usually refers to finding an optimal solution that is feasible for all possible realizations (often represented as scenarios) induced by an uncertainty set. For an extensive overview of theoretical properties and applications, we refer the reader to [6, 8, 17]. Instead of finding the best robust solution for the worst-case scenario, in *regret robustness* [24], the objective is to minimize the difference (*regret*) between the robust solution value and the best objective value we could have achieved if the realization had been known beforehand.

Since these approaches still require the solution to be feasible for all possible scenarios, the concept of *light robustness* [42] weakens the feasibility constraint by only searching for solutions that are feasible and “good enough” in the *nominal case* (e.g., the most likely case). An alternative idea to overcome over-conservatism is to bound the uncertainty set by parameterizing the allowed cumulative deviation from the nominal case, resulting in a *budgeted uncertainty set* [7]. A similar approach is followed for the *optimization problems under controllable uncertainty* [26], where the uncertainty is allowed to be reduced at a given cost.

The concept *adjustable robustness* [5] takes into account that, in many applications, it is possible to adjust a part of the decisions after the uncertainty is unveiled. The idea is to divide the variables into those that must be decided before the uncertainty is unveiled and those that can be decided after its realization. For an overview on adjustable robust optimization, we refer the reader to [53].

An alternative stream of literature addresses network uncertainty through probabilistic resilience guarantees, seeking solutions that remain viable with at least a given probability threshold under arbitrary failures [2, 12]. While such approaches offer

attractive coverage against unforeseen disruptions, they typically require an explicit probabilistic model of the uncertainty, which is a key differentiator regarding our choice of uncertainty modeling.

As previous concepts mainly concentrate on applications in bioinformatics, the focus to date has primarily been on dealing with data inaccuracies. To the best of our knowledge, this is the first work that considers MFDs in the context of classical robustness concepts. This opens up many other application possibilities that are typically represented using flow networks and in which uncertainty plays a role. In public transport, for example, it is now also possible to take demand uncertainties into account in later planning steps, such as vehicle scheduling. In general, this allows robust and, depending on the application, resource-efficient solutions to be achieved despite uncertain data.

3 Minimum Flow Decomposition

Let $G(V, E)$ be a directed acyclic graph (DAG), where $s \in V$ represents the *source* node with no incoming edges and $t \in V$ the *sink (target)* node with no outgoing edges. Let f_{uv} be the corresponding non-negative integer flow value of edge (u, v) for all $(u, v) \in E$, i.e., $f : E \rightarrow \mathbb{N}_{\geq 0}$. Definition 1 provides a formal description of a *flow network*. An example of a flow network is given in Fig. 1a.

Definition 1 (Flow network) The tuple $G(V, E, f)$ is called a *flow network* if for every $v \in V \setminus \{s, t\}$ the conservation of flow

$$\sum_{u:(u,v) \in E} f_{uv} = \sum_{w:(v,w) \in E} f_{vw} \quad (1)$$

is satisfied.

Given such a flow network, it can be decomposed into a set of *s-t-paths*, where each path has an associated positive weight (Fig. 1b). Definition 2 describes how *s-t-paths* form a *k-flow decomposition*.

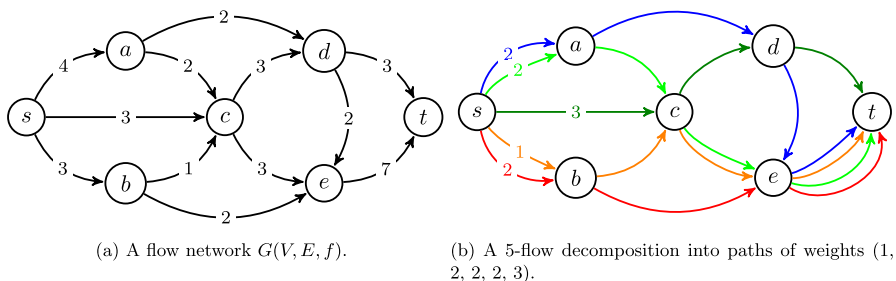


Fig. 1 Example of a flow network and a flow decomposition into 5 *s-t-paths* [14]

Definition 2 (k -flow decomposition) For a given flow network $G(V, E, f)$, a set of s - t -paths $\mathcal{P} = (P_1, \dots, P_k)$ with corresponding positive weights $w = (w_1, \dots, w_k)$ is called k -flow decomposition if

$$\sum_{i: (u,v) \in P_i} w_i = f_{uv} \quad (2)$$

holds for all $(u, v) \in E$.

Remark 3 Let us define $|f|$ of a flow f as the net flow $\sum_{u:(u,t) \in E} f_{ut}$ into the sink node t [20]. As we assume integer flow values, the trivial $|f|$ -flow decomposition with $|f|$ paths of weight one [1] provides an upper bound for k .

Problem 4 (Minimum flow decomposition problem) For a given flow network $G(V, E, f)$, the *minimum flow decomposition problem (MFDP)* is to find a k -flow decomposition $\mathcal{P} = (P_1, \dots, P_k)$ such that k is minimized.

Example 5 provides an illustration of Problem 4, including examples of a given network flow (cf. Definition 1) and a feasible 5-flow decomposition (cf. Definition 2).

Example 5 ([14]) Assume we are given the graph in Fig. 1a with flow values on the corresponding edges. Then, this flow network can be decomposed into the five colored paths P_1, P_2, P_3, P_4, P_5 shown in Fig. 1b with the associated weights $w_1 = 1, w_2 = 2, w_3 = 2, w_4 = 2, w_5 = 3$. Moreover, this is an optimal solution to MFDP with a solution value of $k = 5$.

For a given upper bound $\bar{K}$ of k (for example, $|f|$, cf. Remark 2), Problem 4 can be posed as the corresponding MIP formulation¹.

$$\min_{x,y,w} \sum_{i=1}^{\bar{K}} y_i \quad (3a)$$

$$\text{s.t.} \quad \sum_{v:(s,v) \in E} x_{svi} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (3b)$$

$$\sum_{u:(u,t) \in E} x_{uti} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (3c)$$

$$\sum_{(u,v) \in E} x_{uvi} - \sum_{(v,w) \in E} x_{vwi} = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\}, \quad (3d)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} = f_{uv}, \quad \forall (u, v) \in E, \quad (3e)$$

$$w_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (3f)$$

$$x_{uvi} \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\}, \quad (3g)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (3h)$$

¹ Note that Eq. 3 is formally an MIP only if we linearize constraints Eq. 3e, which can be done using standard linear reformulation techniques (see, e.g., [15])

Formulation Eq. 3 is essentially the formulation in [15]. Here, the binary variable y_i indicates whether the path i with the corresponding weight $w_i \in \mathbb{Z}^+$ is part of the flow decomposition ($y_i = 1$) or not ($y_i = 0$). The variable $x_{uvi} \in \{0, 1\}$ is equal to one if edge $(u, v) \in E$ is part of path $i \in \{1, \dots, \bar{K}\}$. Every path i starts at node s (i.e., has exactly one variable x_{svi} set to 1 for some edge $(s, v) \in E$) and ends at node t (has exactly one variable x_{vti} set to 1, for some edge $(v, t) \in E$). This is enforced by Eqs. 3b and 3c, respectively. Constraint Eq. 3d, together with Eqs. 3b and 3c, characterize a path from s to t . Finally, Eq. 3e enforces that the combined weights w_i of all paths i that traverse the edge (u, v) match the flow $f_{u,v}$. Constraints Eqs. 3f–3h define variable domains. Note that one can straightforwardly add symmetry-breaking constraints to Eq. 7, e.g., by imposing ordering constraints on the variables y and w .

The MFDP can be generalized by relaxing constraints Eq. 3e, allowing the summed path weights of each edge (u, v) to lie in a range $[f_{uv}^l, f_{uv}^u]$ instead of matching a flow value f_{uv} . Consequently, the underlying graph is now an *inexact flow network* [51], meaning that we have a lower bound f_{uv}^l and an upper bound flow value f_{uv}^u corresponding to each edge $(u, v) \in E$. The resulting problem, the minimum inexact flow decomposition problem (MIFDP), is summarized in Problem 6. Note that both MFDP and its generalization, MIFDP, are NP-hard [14].

Problem 6 (Minimum inexact flow decomposition problem [51]) For a given inexact flow network $G(V, E, f^l, f^u)$, the minimum inexact flow decomposition problem (MIFDP) is to find a set of s - t -paths $\mathcal{P} = (P_1, \dots, P_k)$ with minimum cardinality and associated positive weights $w = (w_1, \dots, w_k)$ such that

$$f_{uv}^l \leq \sum_{i:(u,v) \in P_i} w_i \leq f_{uv}^u \quad (4)$$

holds for all $(u, v) \in E$. We obtain its corresponding MIP formulation by replacing constraints Eq. 3e in Eq. 3 with Eq. 5

$$\begin{aligned} \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\geq f_{uv}^l, \quad \forall (u, v) \in E, \\ \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\leq f_{uv}^u, \quad \forall (u, v) \in E. \end{aligned} \quad (5)$$

4 Minimum Weighted Inexact Flow Decomposition Problem

In this section, we introduce a generalized version of the MIFDP by minimizing the weighted sum of the associated weights $a_w \sum_{i=1}^{|\mathcal{P}|} w_i$ in addition to the number of paths $a_y |\mathcal{P}|$. Their presence in the objective function with corresponding weights $a_y, a_w \in \mathbb{R}_{\geq 0}$ motivates the analysis of the subproblems resulting from different weight combinations. While $a_y > 0, a_w = 0$ corresponds to the MIFDP, we show how $a_y = 0, a_w > 0$ can be solved in polynomial time. Moreover, polynomial time can also be achieved for $a_y > 0, a_w = 0$ if the upper bounds on the flow values are neglected.

4.1 General Formulation

Using the MIFDP as a starting point, the weight variables w no longer necessarily match the flow values since the summed path weights of each edge (u, v) lie in a range $[f_{uv}^l, f_{uv}^u]$ instead. Depending on the magnitude of the upper bounds in f^u , the entries of w can take on large values, even if smaller ones would be sufficient to cover the corresponding lower bounds. Therefore, it is reasonable to extend the objective function by the sum of all weights, as in many related applications, the weights are desired to be as small as possible, e.g., if they correspond to a vehicle's capacity. The resulting problem is formalized next.

Problem 7 (Minimum weighted inexact flow decomposition problem) For a given inexact flow network $G(V, E, f^l, f^u)$, the *minimum weighted inexact flow decomposition problem (MWIFDP)* is to find a set of s - t -paths $\mathcal{P} = (P_1, \dots, P_k)$ with associated positive weights $w = (w_1, \dots, w_k)$ such that

$$f_{uv}^l \leq \sum_{i:(u,v) \in P_i} w_i \leq f_{uv}^u$$

holds for all $(u, v) \in E$ and the weighted sum

$$a_y |\mathcal{P}| + a_w \sum_{i=1}^{|\mathcal{P}|} w_i, \quad (6)$$

with weights $a_y, a_w \in \mathbb{R}_{\geq 0}$ is minimized.

For a given upper bound $\bar{K}$ of k , the corresponding MIP formulation for the MWIFDP is given by

$$\min_{x, y, w} \sum_{i=1}^{\bar{K}} (a_y y_i + a_w w_i) \quad (7a)$$

$$\text{s.t.} \quad \sum_{v:(s,v) \in E} x_{svi} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (7b)$$

$$\sum_{u:(u,t) \in E} x_{uti} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (7c)$$

$$\sum_{(u,v) \in E} x_{uvi} - \sum_{(v,w) \in E} x_{vwi} = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\}, \quad (7d)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} \geq f_{uv}^l, \quad \forall (u, v) \in E, \quad (7e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} \leq f_{uv}^u, \quad \forall (u, v) \in E, \quad (7f)$$

$$w_i \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \quad (7g)$$

$$x_{uvi} \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\}, \quad (7h)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (7i)$$

Corollary 8 For $a_y > 0$ and $a_w = 0$, MWIFDP is NP-hard.

Notice that MWIFDP reduces to MIFDP for $a_y > 0$ and $a_w = 0$ and is therefore NP-hard. In Lemma 9, we show that this is also the case for $a_y, a_w > 0$.

Lemma 9 For $a_y, a_w > 0$, MWIFDP is NP-hard since it includes the MFDP.

Proof Given an instance $G(V, E, f)$ of MFDP, we can construct an instance $G(V, E, f^l, f^u)$ of MWIFDP using $f_{uv}^l = f_{uv} = f_{uv}^u$ for all $(u, v) \in E$. Then, the summed weights of the paths are equal to the summed flow values of all outgoing edges of s (the complete s - t -flow value) and build both a lower and upper bound on $\sum_{i=1}^K w_i$. Consequently, finding a minimum flow decomposition, which is NP-hard [47], minimizes the objective. $\square$

Counterintuitively, we show that the MWIFDP is not NP-hard when $a_y = 0$ and $a_w > 0$. To show this, we first revisit the *minimum-cost flow problem* [35] in capacitated networks.

Definition 10 (Minimum-cost flow problem [35]) Given a tuple $G(V, E, d, c)$, where $G(V, E)$ is a digraph, $d(v) \in \mathbb{Z}$ represents the demand ($d(v) \leq 0$) or the supply ($d(v) > 0$) of node $v \in V$ and $c(u, v) \geq 0$ is the cost associated with every edge $(u, v) \in E$. The *minimum-cost flow problem* (MCFP) is to find a flow $f \in \mathbb{N}_{\geq 0}^{|E|}$ with minimum cost $\sum_{(u,v) \in E} c(u, v) \cdot f_{uv}$ such that

$$\sum_{u:(u,v) \in E} f_{uv} - \sum_{w:(v,w) \in E} f_{vw} = d(v) \quad (8)$$

is satisfied for all $v \in V$. In capacitated networks, we have $G(V, E, d, c, \underline{f}, \bar{f})$ with lower $\underline{f}_{uv}$ and upper bounds $\bar{f}_{uv}$ on the flow f_{uv} associated with each edge $(u, v) \in E$, which also have to be fulfilled.

Lemma 11 For $a_y = 0$ and $a_w > 0$, MWIFDP can be solved in polynomial time.

Proof This particular case can be reduced to the minimum-cost flow problem in capacitated networks, which can be solved in polynomial time [35, 49, 55].

For a given inexact flow network $G(V, E, f^l, f^u)$, we show that solving the MWIFDP with $a_y = 0$ and $a_w > 0$ corresponds to finding a minimal cost flow in the following graph.

For $G'(V', E', d, c, \underline{f}, \bar{f})$, let $V' = V \cup \{s'\} \cup \{t'\}$ be the node set with a super-source s' and super-sink t' and $E' = E \cup \{(s', s), (t, t'), (s', t')\}$ the corresponding edges.

We set the supply and demand of s' and t' equal to the summed upper bounds on the flow values $\sum_{(u,v) \in E} f_{uv}^u =: M$, i.e., $d(s') = M = -d(t')$, while we have $d(v) = 0$ for all other nodes $v \in V$. For the cost function c , we have $c(s', s) = 1$ and $c(e) = 0$ for all $e \in E' \setminus \{(s', s)\}$, and the lower and upper bounds, $\underline{f}, \bar{f}$, on the flow are defined as $(\underline{f}_{uv}, \bar{f}_{uv}) := (f_{uv}^l, f_{uv}^u)$ for $(u, v) \in E$ and $(\underline{f}_{uv}, \bar{f}_{uv}) := (0, M)$ for $(u, v) \in \{(s', s), (t, t'), (s', t')\}$.

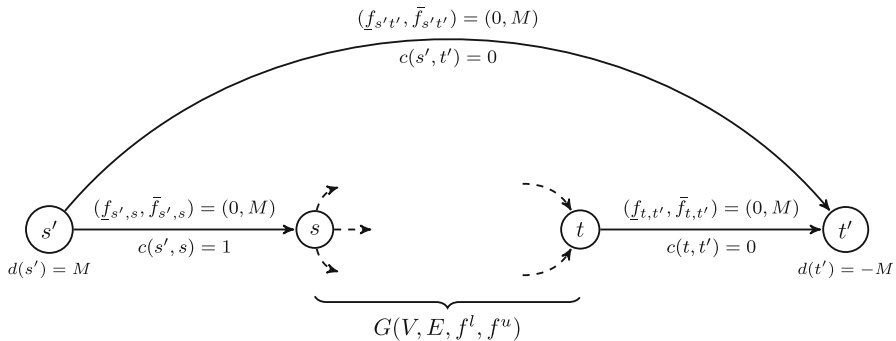


Fig. 2 Illustration of transformed graph $G'(V', E', d, c, \underline{f}, \bar{f})$ for the MCFP to solve the MWIFDP for $a_y = 0$ and $a_w > 0$ in $G(V, E, f^l, f^u)$

In other words, we extend the original graph $G(V, E, f^l, f^u)$ so that all flow units required to fulfill the constraints on the bounds f^l and f^u have to use edge (s', s) at a cost of 1 each. The remaining flow can use the edge (s', t') with a cost of 0. An illustration of $G'(V', E', d, c, \underline{f}, \bar{f})$ can be found in Fig. 2.

If we now consider a minimum-cost flow f^m in $G'(V', E', d, c, \underline{f}, \bar{f})$, we can construct a feasible flow f for the MWIFDP in the original graph $G(V, E, f^l, f^u)$ by removing edges (s', s) , (t, t') , (s', t') with corresponding flow values. That means we have a flow $f \in \mathbb{N}_{\geq 0}^{|E|}$ with $f_{uv} := f_{uv}^m$ for $(u, v) \in E$ that is a feasible solution for MWIFDP by fulfilling the same flow and bound constraints. Since each flow unit in f traverses edge (s', s) with cost 1, the total flow value equals the flow cost $c(f^m) := \sum_{(u,v) \in E'} c(u, v) \cdot f_{uv}^m = f_{s's}^m$ and is therefore minimal.

Finally, notice that the number of s - t -paths can be arbitrarily large without affecting the objective function. In turn, we can trivially decompose f into $c(f^m)$ s - t -paths each with weight 1, which minimizes $a_w \sum_{i=1}^{\bar{K}} w_i = a_w c(f^m)$ in Eq. 7. $\square$

4.2 Formulation Without Upper Bounds

For the remainder of this section, we neglect the upper bounds on the flow, i.e., we consider Eq. 7 without constraints Eq. 7f. This allows us to develop additional complexity results that are useful for both the MWIFDP and the strictly robust counterparts of the different variants to be presented in the next section.

Lemma 12 *For $a_y = 0$ and $a_w > 0$, the MWIFDP without upper bounds can be solved in polynomial time.*

Proof We follow the proof of Lemma 21, but since we have no upper bounds, we set them to ∞ for the MCFP, i.e., we have $(\underline{f}_{uv}, \bar{f}_{uv}) = (f_{uv}^l, \infty)$. Consequently, the same reasoning also applies here: Since the flow constraints have to be satisfied and the lower bound for each $(u, v) \in E$ is $\underline{f}_{uv}$, each edge (with $\underline{f}_{uv} > 0$) is covered by one or more paths with a summed weight of at least $\underline{f}_{uv}$. Since each s - t -path is an s' - t' path that uses edge (s', s) with $c(s', s) = 1$, the summed weights of all s - t -paths are minimized. $\square$

If we consider the alternative case with $a_y > 0$ and $a_w = 0$, i.e., the MIFDP without upper bounds, we obtain a different complexity result than in Sect. 4.1 (Corollary 8).

Lemma 13 *For $a_y > 0$ and $a_w = 0$, the MWIFDP without upper bounds can be solved in polynomial time.*

Proof Due to the fact that the weights w_i can be arbitrarily large without affecting the objective function, we can choose

$$w_i := \sum_{(u,v) \in E} f_{uv}^l$$

for $i = 1, \dots, \overline{K}$ and the problem reduces to finding a minimal number of s - t -paths covering all edges with $f_{uv}^l > 0$.

Once again, we can use the proof of Lemma 11 to show that this corresponds to finding a minimal cost flow. However, there is a modification that is required for the lower bounds f , as we only want to cover the edges that have a positive flow. Therefore, it is sufficient to set 1 as the lower bound for these. Consequently, we have $(\underline{f}_{uv}, \tilde{f}_{uv}) := (1, \infty)$ for all $(u, v) \in E$ with $f_{uv}^l > 0$ and $(\underline{f}_{uv}, \tilde{f}_{uv}) := (0, \infty)$ for $(u, v) \in \{(s', s), (t, t'), (s', t')\} \cup \{(u, v) \in E : f_{uv}^l = 0\}$.

Considering an optimal minimum-cost flow in this modified graph, each relevant edge is covered by at least one s' - t' -path (and thus s - t -path), while the first edge of each $s' - t'$ path has cost 1. Consequently, we minimize the number of s - t -paths covering all relevant edges. $\square$

Lastly, we show that, although both previous cases can be solved in polynomial time, this is not the case for $a_y, a_w > 0$.

Lemma 14 *For $a_y, a_w > 0$, MWIFDP without upper bounds is NP-hard.*

Proof We follow the proof of Proposition 2 in [47]. Therefore, we show that the NP-hard 3-PARTITION problem [18] can be reduced to MWIFDP without upper bounds.

An instance (A, s) of 3-PARTITION consists of $3b$ Elements $a \in A$, where each element a is associated with a size $s(a) \in \mathbb{N}_{\geq 1}$ with $\sum_{a \in A} s(a) = bB$ and $\frac{B}{4} < s(a) < \frac{B}{3}$. To decide whether there exists a partition of A into b disjoint subsets of size B is known to be NP-hard.

Now, we construct an instance $G(V, E, f^l)$ of MWIFDP without upper bounds associated with the instance (A, s) of 3-PARTITION. We have $V = \{s, o, t\}$, $3b$ parallel edges e'_a of the form (s, o) corresponding to element $a \in A$ with $f_{e'_a}^l := s(a)$ and b parallel edges e''_i of the form (o, t) with $i \in \{1, \dots, b\}$ and $f_{e''_i}^l := B$. Evidently, we need at least $3b$ paths to cover the lower bounds of all parallel edges, i.e.,

$$a_y |\mathcal{P}| \geq a_y \times 3b.$$

Since the sum of the lower bounds of the outgoing edges of s is $\sum_{a \in A} f_{e'_a}^l = \sum_{a \in A} s(a) = bB$, the summed weights of the paths must therefore be at least bB , i.e.,

$$a_w \sum_{i=1}^{|\mathcal{P}|} w_i \geq a_w \times bB.$$

Following the same reasoning as in [47], there is a partition of A into b disjoint subsets with size B if and only if there is a $3b$ -flow decomposition into paths with a total weight of bB . This means that the optimal objective value of MWIFDP without upper bounds, for instance, $G(V, E, f^l)$, is equal to

$$a_y |\mathcal{P}| + a_w \sum_{i=1}^{|\mathcal{P}|} w_i = a_y \times 3b + a_w \times bB. \quad \square$$

5 Robust Minimum Flow Decomposition

When considering uncertainty and (inexact) flow networks in general, it is natural that the flow values are the source of uncertainty. To provide formulations that can take into account the uncertainty associated with MFD flow values, we make use of two different notions of robustness, which, for the sake of completeness, we briefly describe next.

5.1 Strict and Adjustable Robustness

Let $\mathcal{U} \subset \mathbb{R}^N$ be an uncertainty set, i.e., a set of possible realizations for the uncertain parameter $\xi \in \mathcal{U}$. For a given optimization under uncertainty problem of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & F(x, \xi) \leq 0, \\ & x \in \mathbb{R}^n, \end{aligned} \quad (9)$$

with $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ and $F(\cdot, \xi) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the *strictly robust counterpart* [50] is defined by

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & F(x, \xi) \leq 0, \quad \forall \xi \in \mathcal{U}, \\ & x \in \mathbb{R}^n. \end{aligned} \quad (10)$$

An optimal solution to Eq. 10 is called a *strictly robust solution*, i.e., we search for the best solution that is feasible for all scenarios $\xi \in \mathcal{U}$.

Strict robustness is often too conservative, especially regarding the requirement that the solution must be feasible for all scenarios within the uncertainty set. As a result, the literature is rich in alternative forms of robustness, which have also been investigated [8, 17].

In some cases, it is reasonable to assume that part of the decisions, the so-called *wait-and-see decisions*, can be made after the scenario ξ is revealed, while the *here-and-now decisions* have to be decided before the scenario ξ is observed. More formally, we can distinguish between *non-adjustable variables* $u \in \mathbb{R}^p$ (here-and-now decisions, as in the strict robust case) and *adjustable variables* $v \in \mathbb{R}^q$ (wait-and-see decisions), whose

values may depend on the realization ξ . This concept is called *adjustable robustness* [5], and for $x = (u, v) \in \mathbb{R}^n$, the *adjustable robust counterpart* of Eq. 10 is defined as

$$\begin{aligned} \min_{u \in S_u} \quad & \min_{v \in \mathbb{R}^q} f(u, v) \\ \text{s.t.} \quad & F(u, v, \xi) \leq 0, \quad \forall \xi \in \mathcal{U}, \end{aligned} \quad (11)$$

with $p + q = n$ and $S_u = \{u \in \mathbb{R}^p : \forall \xi \in \mathcal{U}, \exists v \in \mathbb{R}^q : F(u, v, \xi) \leq 0\}$ representing general feasibility conditions for the choice of u . This means that we make the decisions represented by the non-adjustable variables u , observe ξ , and then make decisions associated with the adjustable variables v depending on the scenario ξ and u . An optimal solution to Eq. 11 is called an *adjustable robust solution*. Note that for $p = n$, i.e., $x = u \in \mathbb{R}^n$, the adjustable robust counterpart Eq. 11 reduces to the strictly robust counterpart Eq. 10.

Using interval-wise uncertainty, we can define the scenarios according to the deviation from a nominal scenario $\hat{\xi}$ (e.g., the expected value or the most likely scenario). This means that the uncertainty set is of the form

$$\mathcal{U}^\square = \left\{ \xi \in \mathbb{R}_{\geq 0}^N : \xi_i \in [\hat{\xi}_i - \bar{\delta}_i, \hat{\xi}_i + \bar{\delta}_i], \forall i \in \{1, \dots, N\} \right\}, \quad (12)$$

for given $\bar{\delta}, \bar{\delta} \in \mathbb{R}_{\geq 0}^N$, where ξ_i corresponds to the i -th entry of scenario ξ .

To prevent the uncertainty set from leading to over-conservative solutions, the cumulative deviation from the nominal scenario can be bounded by the uncertainty budget $\Gamma \in \mathbb{R}_{\geq 0}$. More precisely, $\mathcal{U}(\Gamma)$ is of the form

$$\mathcal{U}(\Gamma) = \left\{ \xi \in \mathbb{R}_{\geq 0}^N : \xi_i \in [\hat{\xi}_i - \bar{\delta}_i, \hat{\xi}_i + \bar{\delta}_i], \forall i \in \{1, \dots, N\}; \sum_{j=1}^N |\xi_j - \hat{\xi}_j| \leq \Gamma \right\}. \quad (13)$$

This is known as *cardinality-constraint uncertainty set* or *budgeted uncertainty set* [7]; hereinafter, we refer to it as Γ -uncertainty set.

5.2 Applying the Notion of Strict Robustness to the MWIFDP

If we apply the notion of strict robustness to the (exact) MFDP with uncertain flows, the strictly robust counterpart Eq. 10 becomes infeasible if there are at least two different scenarios $f' \neq f'' \in \mathcal{U}$ since there is at least one (u, v) with $f'_{uv} \neq f''_{uv}$ and constraints Eq. 3e cannot be fulfilled.

In general, this is not the case for the MIFDP since the summed path weights of each edge (u, v) must lie within a range $[f'_{uv}, f''_{uv}]$ instead of matching exactly a flow value f_{uv} . Consequently, it is possible that different scenarios $(f^l, f^u)' \neq (f^l, f^u)'' \in \mathcal{U}$ form a feasible combination of constraints in Eq. 5.

Remark 15 Since we are dealing with non-negative integer flows, non-integer bounds on the flow values can also be reduced to integers by rounding up or down. Consequently, it is sufficient to consider a discrete uncertainty set consisting of values in

$\mathbb{N}_{\geq 0}$ since any other uncertainty set for this problem can be reduced to its discrete version. Therefore, we set

$$\mathcal{U} = \left\{ (f^{l,\xi}, f^{u,\xi}) \in \left(\mathbb{N}_{\geq 0}^{|E|} \right)^2 : \xi = 1, \dots, N \right\}, \quad (14)$$

i.e., we have $|\mathcal{U}|$ scenarios, each scenario ξ consisting of a lower $f_e^{l,\xi}$ and an upper bound $f_e^{u,\xi}$ for all edges $e \in E$ (w.l.o.g., we assume $f_e^{l,\xi} \leq f_e^{u,\xi}$).

Having the uncertainty set $\mathcal{U}$ described as Eq. 14 significantly enhances tractability in our setting since we can show that the uncertainty set can be equivalently replaced by a singleton containing the *worst-case* realization $\tilde{f}$. This is stated in Lemma 16.

Lemma 16 *For the strictly robust counterpart of MWIFDP, the uncertainty set $\mathcal{U}$ in Eq. 14 can be reduced to $\tilde{\mathcal{U}} = \{\tilde{f}\}$, i.e., there exists one worst-case scenario $\tilde{f} = (f^{l,\tilde{\xi}}, f^{u,\tilde{\xi}})$.*

Proof Concerning the strictly robust counterpart of MWIFDP, only constraints Eqs. 7c–7f are affected by the uncertainty. More precisely, we have

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} \geq f_{uv}^{l,\xi} \quad \forall (u, v) \in E, \quad \forall (f^{l,\xi}, f^{u,\xi}) \in \mathcal{U}, \quad (15)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} \leq f_{uv}^{u,\xi} \quad \forall (u, v) \in E, \quad \forall (f^{l,\xi}, f^{u,\xi}) \in \mathcal{U}. \quad (16)$$

Every feasible solution to MWIFDP with $f_{uv}^l = \max_{\xi \in \{1, \dots, N\}} f_{uv}^{l,\xi}$ and $f_{uv}^u = \min_{\xi \in \{1, \dots, N\}} f_{uv}^{u,\xi}$ for all $(u, v) \in E$ is feasible for all other scenarios $(f^{l,\xi}, f^{u,\xi}) \in \mathcal{U}$. Consequently, for $(u, v) \in E$, we can write

$$\begin{aligned} \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\geq f_{uv}^{l,\xi} && \forall (f^{l,\xi}, f^{u,\xi}) \in \mathcal{U}, \\ \iff \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\geq \max_{\xi \in \{1, \dots, N\}} f_{uv}^{l,\xi} =: f_{uv}^{l,\tilde{\xi}} \end{aligned}$$

and

$$\begin{aligned} \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\leq f_{uv}^{u,\xi} && \forall (f^{l,\xi}, f^{u,\xi}) \in \mathcal{U}, \\ \iff \sum_{i \in \{1, \dots, \bar{K}\}} w_i x_{uvi} &\leq \min_{\xi \in \{1, \dots, N\}} f_{uv}^{u,\xi} =: f_{uv}^{u,\tilde{\xi}}. \end{aligned}$$

This means that we can project the uncertainty set $\mathcal{U}$ to two axes per edge associated with the lower and upper bounds and determine the maximum and minimum, respectively. Consequently, only the worst-case scenario $\tilde{f} = (f_{uv}^{l,\tilde{\xi}}, f_{uv}^{u,\tilde{\xi}})_{(u,v) \in E}$ can be considered instead of the entire uncertainty set $\mathcal{U}$. $\square$

Remark 17 Note that $\tilde{f} \notin \mathcal{U}$ is possible. This is an artifact of the static (as opposed to adjustable) setting of our strict robustness, where robustness is consequently considered constraint-wise [3].

For a general discrete uncertainty set $\mathcal{U}$ as defined in Eq. 14, it is not immediately clear how to identify a worst-case scenario $\tilde{f}$. For an exponentially large uncertainty set, computing a worst-case scenario can be of exponential complexity. However, for $\mathcal{U}^\square$ in Eq. 12 and for $\mathcal{U}(\Gamma)$ in Eq. 13, $\tilde{f}$ can be identified in linear time. Thus, we show based on the results from Sect. 4 that strictly robust solutions can be computed in polynomial time.

First, consider an interval-wise uncertainty set

$$\mathcal{U}^\square = \left\{ (f^l, f^u) \in \left(\mathbb{N}_{\geq 0}^{|E|} \right)^2 : f_{uv}^l \in [\underline{f}_{uv}^l, \bar{f}_{uv}^l], f_{uv}^u \in [\underline{f}_{uv}^u, \bar{f}_{uv}^u], \forall (u, v) \in E \right\}, \quad (17)$$

with given $\underline{f}^l, \bar{f}^l, \underline{f}^u, \bar{f}^u \in \mathbb{N}_{\geq 0}^{|E|}$. Note that since the flow values are integers, Eq. 14 contains Eq. 17 and can be constructed by enumerating all elements in an interval-wise uncertainty set.

Lemma 18 *For the strictly robust counterpart of MWIFDP, the uncertainty set $\mathcal{U}^\square$ as defined in Eq. 17 can be reduced to $\mathcal{U} = \{(f^l, f^u)\}$ with $f_{uv}^l = \bar{f}_{uv}^l$ and $f_{uv}^u = \underline{f}_{uv}^u$ for all $(u, v) \in E$.*

Proof It follows directly from the proof of Lemma 16 that every optimal solution to MWIFDP with $f_{uv}^l = \bar{f}_{uv}^l$ (maximum lower bound) and $f_{uv}^u = \underline{f}_{uv}^u$ (minimum upper bound) for all $(u, v) \in E$ is a strictly robust solution. $\square$

Now, if we restrict $\mathcal{U}^\square$ in Eq. 17 to a Γ -uncertainty set $\mathcal{U}(\Gamma)$, i.e., $\mathcal{U}(\Gamma)$ must fulfill

$$\sum_{(u,v) \in E} |f_{uv}^l - \hat{f}_{uv}^l| + |f_{uv}^u - \hat{f}_{uv}^u| \leq \Gamma \quad (18)$$

concerning a nominal scenario $(\hat{f}^l, \hat{f}^u)$ with $\hat{f}_{uv}^l \in [\underline{f}_{uv}^l, \bar{f}_{uv}^l]$ and $\hat{f}_{uv}^u \in [\underline{f}_{uv}^u, \bar{f}_{uv}^u]$ for $(u, v) \in E$, the resulting worst-case scenario is only slightly different.

Lemma 19 *For the strictly robust counterpart of MWIFDP, the Γ -uncertainty set $\mathcal{U}(\Gamma)$ can be reduced to $\mathcal{U}(\Gamma) = \{(f^l, f^u)\}$ with $f_{uv}^l = \min\{\hat{f}_{uv}^l + \Gamma, \bar{f}_{uv}^l\}$ and $f_{uv}^u = \max\{\hat{f}_{uv}^u - \Gamma, \underline{f}_{uv}^u\}$ for all $(u, v) \in E$.*

Proof We use the proof of Lemma 16 again, but this time, we have to take the limited budget Γ into account. Since the robust solution has to be feasible for any realization, a scenario that covers all worst-case scenarios is $(f_{uv}^l, f_{uv}^u) = (\min\{\hat{f}_{uv}^l +$

$\Gamma, \bar{f}_{uv}^l\}$, $\max\{\hat{f}_{uv}^u - \Gamma, \underline{f}_{uv}^u\}$ for all $(u, v) \in E$. Note that this particular scenario is not necessarily part of $\mathcal{U}(\Gamma)$. $\square$

Remark 20 Other alternative geometries for the uncertainty set $\mathcal{U}$ can be considered in this setting. Following Lemma 16, the tractability of the strict robust versions of MFDP and MIFDP hinges on being able to identify the worst-case realization $\hat{f}$ efficiently.

Since we have shown in Lemma 16 that any discrete uncertainty set of the form Eq. 14 can be reduced to a single worst-case scenario, the results from Sect. 4 also apply to the strictly robust counterpart of MWIFDP.

Corollary 21 For $a_y = 0$ and $a_w > 0$, the strictly robust counterpart of MWIFDP with uncertainty sets $\mathcal{U}^\square$ and $\mathcal{U}(\Gamma)$ can be solved in polynomial time.

Proof Using the same reasoning as before, we can reduce the uncertain optimization problem of MWIFDP to solving MWIFDP for the respective worst-case scenario for $\mathcal{U}^\square$ and $\mathcal{U}(\Gamma)$. Consequently, we use the procedure in Lemma 11 with $(\underline{f}_{uv}, \bar{f}_{uv}) = (\bar{f}_{uv}^l, \underline{f}_{uv}^u)$ and $(\underline{f}_{uv}, \bar{f}_{uv}) = (\min\{\hat{f}_{uv}^l + \Gamma, \bar{f}_{uv}^l\}, \max\{\hat{f}_{uv}^u - \Gamma, \underline{f}_{uv}^u\})$ for all $(u, v) \in E$ regarding the uncertainty sets $\mathcal{U}^\square$ and $\mathcal{U}(\Gamma)$, respectively. $\square$

If we neglect the upper bounds on the flow, the uncertainty sets $\mathcal{U}^\square$ and $\mathcal{U}(\Gamma)$ simplify further. Now, let $\mathcal{U}^\square$ be an interval-wise uncertainty set with

$$\mathcal{U}^\square = \left\{ f^l \in \mathbb{N}_{\geq 0}^{|E|} : f_{uv}^l \in [\underline{f}_{uv}^l, \bar{f}_{uv}^l], \forall (u, v) \in E \right\}, \quad (19)$$

for given $\underline{f}^l, \bar{f}^l \in \mathbb{N}_{\geq 0}^{|E|}$ and $\mathcal{U}(\Gamma)$ the corresponding Γ -uncertainty set with

$$\mathcal{U}(\Gamma) = \left\{ f^l \in \mathbb{N}_{\geq 0}^{|E|} : f_{uv}^l \in [\underline{f}_{uv}^l, \bar{f}_{uv}^l], \forall (u, v) \in E; \sum_{(u,v) \in E} |f_{uv}^l - \underline{f}_{uv}^l| \leq \Gamma \right\}. \quad (20)$$

for a given uncertainty budget $\Gamma \in \mathbb{R}_{\geq 0}$ and nominal case $\hat{f} = \underline{f}^l$. We can assume w.l.o.g. that $\Gamma \in \mathbb{N}_{\geq 0}$ since $f^l \in \mathbb{N}_{\geq 0}^{|E|}$.

Remark 22 Notice that the uncertainty sets in Eqs. 19 and 20 are discrete sets.

Now, we can adopt the reasoning in Lemmas 18 and 19 for the lower bounds. Consequently, a worst-case scenario in the strictly robust case is $\bar{f}_{uv}^l$ and $\min\{\hat{f}_{uv}^l + \Gamma, \bar{f}_{uv}^l\}$ for all $(u, v) \in E$ regarding $\mathcal{U}^\square$ and $\mathcal{U}(\Gamma)$, respectively. Moreover, one can notice that Corollary 21 also holds for this modified version of MWIFDP without upper bounds.

Corollary 23 For $a_y = 0$ and $a_w > 0$, the strictly robust counterpart of MWIFDP without upper bounds and with uncertainty sets $\mathcal{U}^\square$ in Eq. 19 and $\mathcal{U}(\Gamma)$ in Eq. 20 can be solved in polynomial time.

Proof We follow the proof of Lemma 12. That means we have $(f_{uv}, \bar{f}_{uv}) = (\bar{f}_{uv}^l, \infty)$ concerning uncertainty set $\mathcal{U}^\square$ and $(f_{uv}, \bar{f}_{uv}) = (\min\{\hat{f}_{uv}^l + \Gamma, \bar{f}_{uv}^l\}, \infty)$ concerning $\mathcal{U}(\Gamma)$. $\square$

Corollary 24 For $a_y > 0$ and $a_w = 0$, the strictly robust counterpart of MWIFDP without upper bounds and with uncertainty sets $\mathcal{U}^\square$ in Eq. 19 and $\mathcal{U}(\Gamma)$ in Eq. 20 can be solved in polynomial time.

Proof With the result from Lemma 16, the uncertainty set $\mathcal{U}^\square$ can be reduced to $\{(f^l, f^u)\}$ with $f_{uv}^l = \bar{f}_{uv}^l$ and $f_{uv}^u = \underline{f}_{uv}^u$ for all $(u, v) \in E$. The same applies to the uncertainty set $\mathcal{U}(\Gamma)$, which can be reduced to $\{(f^l, f^u)\}$ with $f_{uv}^l = \min\{\hat{f}_{uv}^l + \Gamma, \bar{f}_{uv}^l\}$ and $f_{uv}^u = \max\{\hat{f}_{uv}^u - \Gamma, \underline{f}_{uv}^u\}$ for all $(u, v) \in E$. Subsequently, we can follow the proof of Lemma 13. $\square$

5.3 Adjustable Robust Flow Decomposition

As we have seen in the previous section, even if we know that the uncertainty set is restricted and only a few edges in each scenario have a positive flow, we still need to set weights that cover all edges that have a positive flow in any scenario. As a result, the number of paths (and the sum of weights) in the solution is much larger than what is actually required for any realized scenario. Moreover, the uncertainty set must allow for a solution that is feasible for all scenarios. To overcome the rigidity of the solutions obtained by the strictly robust model, we make use of adjustable robustness, introduced in Sect. 5.1, to allow part of the decisions to be made after the uncertainty has been observed.

Depending on the application at hand, it may be essential to know the number of paths a priori or to specify a superset of paths from which to choose. In the context of transcript assembly, detaching the number of paths y and each $\text{path}(x)$ from the weights (w) allows for accommodating errors in the measurement of the abundances, which in turn allows for more robust characterization of sequences despite the uncertainty on the actual splice graph.

Similarly, considering the field of transportation, we can have a situation where we need to know beforehand how many vehicles (corresponding to the number of paths y) will be required but can decide on capacities (w) and routes (x) later, as well as a situation where both the routes (x) and the number of drivers (y) must be known in advance, but the vehicles with suitable capacities (w) can be selected depending on the final realization.

As we can make three different types of decisions (number of paths, paths, weights) corresponding to the three different variable types (y, x, w) , this naturally results in two adjustable robust optimization versions of MWIFDP based on Eq. 7. In the first formulation, we have one here-and-now and two wait-and-see decisions, i.e., we first make a decision on the number of paths y and then decide on the paths x and weights w depending on the observed scenario ξ . Since we only have one set of non-adjustable variables, we call it the more adjustable formulation *MA*. Consequently, in the second less adjustable formulation, denoted by *LA*, we first make decisions on the routes (x)

and how many of them (y), and then we decide on the weights (w) depending on the scenario ξ . An overview of both adjustable formulations can be found in Table 1.

For *MA*, the resulting MIP formulation is stated in Eq. 21. The adjustable variables x^ξ and w^ξ now depend on the scenario $\xi \in \mathcal{U}$ (given by the bounds $f^{l,\xi}$ and $f^{u,\xi}$) and, as a consequence, the sum of weights are moved to the constraints Eq. 21b and substituted by variable $\mathcal{W}$ in the objective function.

$$\min_{x,y,w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (21a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \mathcal{U} \quad (21b)$$

$$\sum_{v:(s,v) \in E} x_{svi}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (21c)$$

$$\sum_{u:(u,t) \in E} x_{uti}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (21d)$$

$$\sum_{(u,v) \in E} x_{uvi}^\xi - \sum_{(v,w) \in E} x_{vwi}^\xi = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\}, \xi \in \mathcal{U} \quad (21e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \geq f_{uv}^{l,\xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (21f)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \leq f_{uv}^{u,\xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (21g)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (21h)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (21i)$$

$$x_{uvi}^\xi \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (21j)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (21k)$$

Table 1 Overview of both adjustable formulations

	<i>MA</i> (more adjustable formulation)	<i>LA</i> (less adjustable formulation)
A priori	Number of paths	Number of paths, paths
A posteriori	Paths, weights	Weights
Non-adjustable	y	y, x
Adjustable	x, w	w
Solution method	Algorithm 1	Algorithm 3
Models	MIP formulation Eq. 21, master problem Eq. 22, sub-problem Eq. 23	MIP formulation Eq. 24, master problem Eq. 25, sub-problem Eq. 26
Features	- Results in a lower number of paths - Depending on the scenario, different paths might be used	Results in a smaller pool of paths covering all scenarios

The formulation for LA is similar, with the key difference being that only the variables w^ξ are adjustable. The complete LA formulation can be found in Eq. 24 in Appendix A.2.

Lemma 25 *The objective value of the optimal solution to MA in Eq. 21 is less than or equal to the objective value of the optimal solution to LA in Eq. 24.*

Proof We show that all feasible solutions to LA in Eq. 24 are also feasible to MA in Eq. 21 and have the same objective value. Let $(\ddot{y}, \ddot{x}, \ddot{w})$ be a feasible solution to LA with

$$(\ddot{y}, \ddot{x}, \ddot{w}) = (\ddot{y}_i, \ddot{x}_{uvi}, \ddot{w}_i^\xi)_{(u,v) \in E, i \in \{1, \dots, \overline{K}\}, \xi \in \mathcal{U}}.$$

Since both formulations Eqs. 24 and 21 differ only in the variable x (Eqs. 24j and 21j), we construct a solution $(\dot{y}, \dot{x}, \dot{w})$ with

$$(\dot{y}, \dot{x}, \dot{w}) = (\dot{y}_i, \dot{x}_{uvi}^\xi, \dot{w}_i^\xi)_{(u,v) \in E, i \in \{1, \dots, \overline{K}\}, \xi \in \mathcal{U}}$$

to MA as follows. We set

$$\begin{aligned} \dot{y}_i &:= \ddot{y}_i, & \forall i \in \{1, \dots, \overline{K}\}, \\ \dot{w}_i^\xi &:= \ddot{w}_i^\xi, & \forall i \in \{1, \dots, \overline{K}\}, \xi \in \mathcal{U}, \\ \dot{x}_{uvi}^\xi &:= \ddot{x}_{uvi}, & \forall (u, v) \in E, i \in \{1, \dots, \overline{K}\}, \xi \in \mathcal{U}, \end{aligned}$$

whereby the latter is feasible because $\ddot{x}$ satisfies constraints Eqs. 24c–24g for all scenarios $\xi \in \mathcal{U}$, and thus, $\dot{x}$ also fulfills constraints Eqs. 21c–21g. It is easy to see that the objective value has not changed, which completes the proof. Notice that this does not apply vice versa, as here the variables x depend on scenario $\xi \in \mathcal{U}$ and are, therefore, generally not feasible for all scenarios. $\square$

Remark 26 Naturally, Eqs. 21 and 24 reduce to the MWIFDP formulation in Eq. 7 if we have a single scenario, but unlike for strict robustness, we need to consider more than one scenario in the general case. Moreover, in contrast to strict robustness, an edge $e \in E$ for which the intersection of the intervals $[f_e^{l,\xi}, f_e^{u,\xi}]$ of all scenarios ξ is empty no longer automatically implies infeasibility.

Remark 27 Both robustness concepts discussed in this paper offer meaningful coverage against unforeseen realizations not explicitly represented in $\mathcal{U}(\Gamma)$. For strict robustness, [7] show that the probability of a feasible solution being violated by such a realization decreases exponentially in Γ , which can thus be calibrated to provide a controllable confidence level. For adjustable robustness, this coverage naturally extends further, since the adjustable variables can be re-optimized upon observing the actual realization of the uncertainty, and thus the a priori decisions are exposed to out-of-sample risk.

The MIP formulations Eqs. 21 and 24 yield challenging problems with steep, impractical computational requirements. Therefore, to solve the proposed adjustable variants LA and MA , we adapt the *column-and-constraint generation* approach in [54]

to find a minimal subset $\bar{\mathcal{U}}$ of $\mathcal{U}$ that is sufficient to *cover* the adjustable problem for the complete uncertainty set $\mathcal{U}$. Here, we say that $\bar{\mathcal{U}} \subset \mathcal{U}$ *covers* $\mathcal{U}$ if solving the problem for $\bar{\mathcal{U}}$ returns the same solution as solving the problem for $\mathcal{U}$. The adaptation relates to the fact that, while in [54] the uncertainty set $\mathcal{U}$ is assumed to be continuous (with a finite number of extreme points and rays), our uncertainty set is discrete and finite. Nevertheless, the convergence result in [54] still guarantees that the proposed column-and-constraint generation algorithm will converge and return an optimal solution for a set $\bar{\mathcal{U}}$ that covers $\mathcal{U}$.

Therefore, we start by solving formulation Eq. 21 (or Eq. 24) with subset $\bar{\mathcal{U}}$ that includes initially one arbitrary scenario. This is the so-called *master problem*. We use the resulting solution of the non-adjustable variables to fix them in the *sub-problem*, where we then solve the formulation for all scenarios $\xi \in \mathcal{U}$. The scenario ξ^* with the highest solution value or that causes infeasibility is added to $\bar{\mathcal{U}}$ and the steps are repeated until the solution value of the master problem equals the solution value of the sub-problem (in particular, it is feasible for all scenarios). For *MA*, a detailed description of the column-and-constraint generation method can be found in Algorithm 1 and the corresponding formulations of the master problem and sub-problem in Eqs. 22 and 23 in Appendix A.1. If a given solution (y^k) for the master problem leads to an infeasible sub-problem, we know that we need at least one more path. As a consequence, in line 12 of Algorithm 1, we add the corresponding constraint to the master problem Eq. 22. If this is the case in *LA*, we only know that the combination of (y^k) and (x^k) causes infeasibility of the sub-problem. Therefore, we add the constraint

$$\sum_{i:(y^k)_i=0} y_i + \sum_{i:(x^k)_i=0} x_i + \sum_{i:(y^k)_i=1} (1 - y_i) + \sum_{i:(x^k)_i=1} (1 - x_i) \geq 1$$

to the master problem Eq. 25 to eliminate this combination. This can be found in Appendix A.2 in Algorithm 3 with the corresponding formulations of the master problem and sub-problem of *LA* in Eqs. 25 and 26.

Even though it may be necessary in theory to consider all scenarios in the master problem, a much smaller number is sufficient in most cases, as we see in the experimental evaluation in Sect. 6.

To evaluate the aforementioned adjustable formulations, we add a third conservative method, the so-called *naive approach*. Here, we solve the problem for each scenario separately and then add all paths with corresponding weights to a set of paths. If solutions for different scenarios use the same path, we add the path only once. To calculate $\mathcal{W}$, we use the maximum of the summed weights of the scenarios. Consequently, this is a heuristic solution for *LA*, giving us an upper bound on *LA* and, thus, also on *MA*.

6 Computational Study

In this section, we experimentally evaluate both adjustable robust optimization versions of the MWIFDP introduced in the previous Sect. 5.3 to quantify the benefit of adaptability when taking into account the uncertainty. Moreover, we show that both *MA* and *LA* considerably outperform the naive approach, in which we solve the prob-

Algorithm 1 Column-and-constraint generation method for *MA*.

```

1: given a scenario set  $\mathcal{U}$  and threshold  $\epsilon$ 
2: set  $LB = -\infty$ ,  $UB = \infty$ ,  $k = 1$  and initialize  $\bar{\mathcal{U}}$  with one scenario from  $\mathcal{U}$ 
3: while  $UB - LB > \epsilon$  do
4:   solve master problem Eq. 22, get optimal solution  $(y^k, x^k, w^k, \mathcal{W}^k)$ 
5:   update lower bound  $LB = \sum_{i=1}^{\bar{K}} (y^k)_i + \mathcal{W}^k$ 
6:   solve sub-problem Eq. 23 with input  $(y^k)$ 
7:   if we obtain an optimal solution  $(\bar{x}^k, \bar{w}^k, \bar{\mathcal{W}}^k)$  then
8:     let  $\xi^k$  be the worst-case scenario causing  $\bar{\mathcal{W}}^k$ 
9:     update upper bound  $UB = \sum_{i=1}^{\bar{K}} (y^k)_i + \bar{\mathcal{W}}^k$ 
10:  else
11:    let  $\xi^k$  be the first scenario causing infeasibility
12:    add  $\sum_{i=1}^{\bar{K}} y_i \geq \sum_{i=1}^{\bar{K}} (y^k)_i + 1$  to constraints of master problem Eq. 22
13:  end if
14:  add  $\xi^k$  to  $\bar{\mathcal{U}}$ 
15:   $k = k + 1$ 
16: end while
17: return  $(y^k, \bar{x}^k, \bar{w}^k)$ 

```

lem for each scenario separately. As described in the previous section, we use the column-and-constraint generation methods in Algorithm 1 and Algorithm 3 as solution methods. The code and data generated are available at www.github.com/gamma-opt/robust-flow-decomposition.

6.1 Data and Scenarios

Due to this being the first work to consider the MWIFDP and additionally adjustable robustness, we could not find instances in the literature that we could use directly for our purposes. Therefore, we adapted graph-structured instances from the literature that are related to our work both theoretically and practically. As mentioned in Sect. 1, the most recent references concerning MFD stem from the field of bioinformatics, providing many datasets for the application of RNA transcript assembly. Here, we select two of the instances also used in [15] and [14], both being acyclic splice graphs, one for a human gene and one for a mouse gene. Additionally, we added a benchmark instance for a minimum-cost flow problem from [48] denoted by *gte* and a graph that contains regional railway data of Lower Saxony, a region in northern Germany. The latter comes from the scientific software toolbox LinTim [41], which deals with solving various planning steps in public transport. Since this is not originally an *s-t*-flow network, we modified it by adding 71 auxiliary edges for all missing outgoing edges of *s* and incoming edges of *t*. To complement this, we also used the artificial graph from Example 5 in Example 5, which was also used in [14]. A description of all instances can be found in Table 2 and the corresponding plots in Figs. 1a and 3.

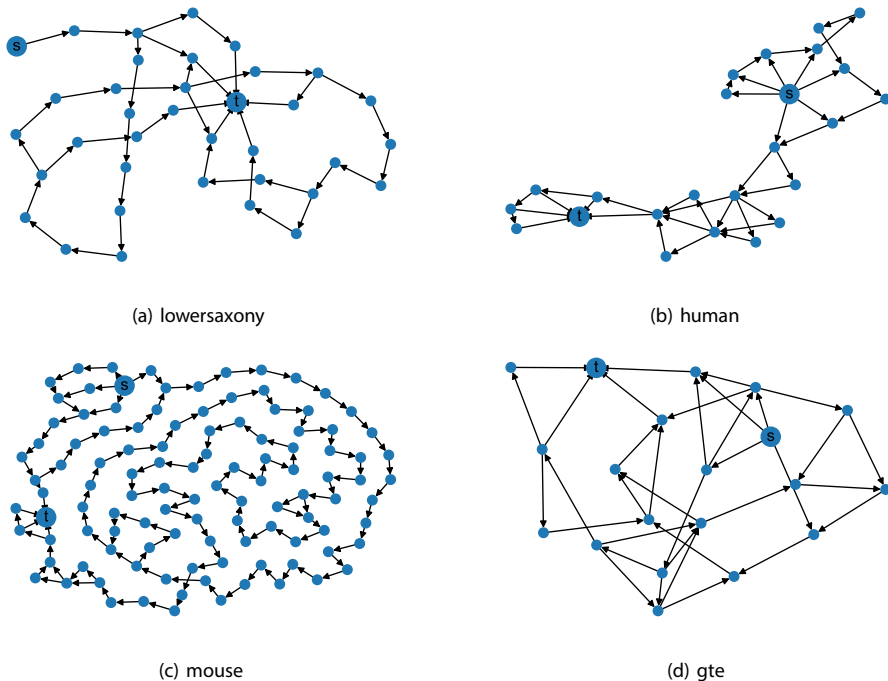
Deriving meaningful scenarios for the aforementioned instances is challenging. Besides the fact that each scenario must be feasible on its own, the difficulty lies in ensuring that the uncertainty set $\mathcal{U}(\Gamma)$ is sufficiently diverse to provide required protection levels while not being so disparate that they become unrealistic or impractical, rendering any robust consideration senseless.

Table 2 Graph properties of the original instances

id	name	# nodes	# edges	origin	plot
1	small	7	12	Example 5	Fig. 1a
2	lowersaxony	36	103	[41]	Fig. 3a
3	human	24	39	[14, 15]	Fig. 3b
4	mouse	111	120	[14, 15]	Fig. 3c
5	gte	49	130	[48]	Fig. 3d

To generate the scenarios that form the uncertainty set, we sample a random subset $\mathcal{P}'$ of s - t -paths of size $p := |\mathcal{P}'|$ in the graph and set the flow value of each edge $e \in E$ to the number of occurrences of e in these s - t -paths. Consequently, $\mathcal{P}'$, with associated path weights all equal to 1, forms a $|\mathcal{P}'|$ -flow decomposition for the constructed flow network with weight $|\mathcal{P}'|$. That means we guarantee feasibility and control the ratio of summed weights to be the number of paths since both are highly correlated and bounded by the subset size $|\mathcal{P}'|$. We then define lower and upper bounds around these flow values for each edge $e \in E$ to construct one scenario. Since we sample a new subset $\mathcal{P}'$ of s - t -paths for each scenario, this promotes variety in the scenarios forming the uncertainty set.

In order to measure and eliminate highly divergent scenarios from the uncertainty set, we first construct a nominal scenario $(f_{uv}^*, f_{uv}^{l,*}, f_{uv}^{u,*}, (u, v) \in E)$ using this method and then only include scenarios whose summed deviations lie within a certain

**Fig. 3** Graph structures of the original instances

range Γ of the nominal scenario. The detailed procedure is outlined in Algorithm 2. Since in our computations, subset sizes $|\mathcal{P}'|$ of value 5, 10, and 20 produce very similar results, we use $|\mathcal{P}'| = 10$ in what follows.

Algorithm 2 Description of scenario generation.

```

1: given digraph  $G(V, E)$ :
2: compute all  $s$ - $t$ -paths  $\mathcal{P}(G)$  and set  $\mathcal{U}(\Gamma) = \emptyset$ ,  $\delta = \frac{p}{2}$ 
3: initialize  $f_{uv}^* = 0$  for all  $(u, v) \in E$  sample a random subset of paths  $\mathcal{P}' \subset \mathcal{P}(G)$  with size  $|\mathcal{P}'| = p$ 
4: update  $f_{uv}^* = f_{uv}^* + 1$  for all  $(u, v) \in P$  for each path  $P \in \mathcal{P}'$ 
5: set  $f_{uv}^{l,*} = f_{uv}^*$  and  $f_{uv}^{u,*} = f_{uv}^* + \delta$  for all  $(u, v) \in E$   $\xi = 1$ 
6: while  $|\mathcal{U}(\Gamma)| < \text{required number of scenarios}$  do
7:   initialize  $f_{uv} = 0$  for all  $(u, v) \in E$ 
8:   sample a random subset of paths  $\mathcal{P}' \subset \mathcal{P}(G)$  with size  $|\mathcal{P}'| = p$ 
9:   update  $f_{uv} = f_{uv} + 1$  for all  $(u, v) \in P$  for each path  $P \in \mathcal{P}'$ 
10:  draw  $f_{uv}^{l,\xi} \in [\max(0, f_{uv} - 2), f_{uv}]$  and set  $f_{uv}^{u,\xi} = f_{uv} + \delta$  for all  $(u, v) \in E$ 
11:  if  $\sum_{e \in E} |f_e^{l,\xi} - f_e^{l,*}| + |f_e^{u,\xi} - f_e^{u,*}| \leq \Gamma' \cdot \sum_{e \in E} (f_e^{l,*} + f_e^{u,*}) := \Gamma$  then
12:    add scenario  $(f^{l,\xi}, f^{u,\xi})$  to uncertainty set  $\mathcal{U}(\Gamma)$ 
13:  end if
14:  update  $\xi = \xi + 1$ 
15: end while
16: return  $\mathcal{U}(\Gamma)$ 

```

To define the range around the flow values for the lower and upper bounds, we use $\delta = \frac{|\mathcal{P}'|}{2}$. Note that f_{uv} is bounded by $|\mathcal{P}'|$. Due to the size of the graphs, f_{uv} is closer to 0 in most cases, which is why we draw the lower bound in the interval $[\max(0, f_{uv} - 2), f_{uv}]$ for all $(u, v) \in E$. Note that all auxiliary edges have a lower bound of zero, i.e., we set the lower bounds $f_{uv}^{l,*}$ and $f_{uv}^{l,\xi}$ for the *lowersaxony* instance to 0 if $u = s$ or $v = t$.

6.2 Experimental Results

As mentioned in the previous section, we set the subset size $|\mathcal{P}'| = 10$. For Γ' , we use 0.1, 0.2, and 0.3, i.e., the allowed summed deviations to the nominal scenario lie within a range of 10 %, 20 %, and 30 %, respectively. As for the uncertainty set sizes, we use $|\mathcal{U}(\Gamma)| \in \{5, 10, 50, 100, 200, 500\}$ since the differences in the results become smaller for the larger sizes. Due to the complexity of the problem, we set a time limit of 24 hours for executing the complete algorithm while ensuring full iterations of the master problem and sub-problem. This means that if the time limit is not reached after solving the master problem (or sub-problem), we do not interrupt the computation of the sub-problem (respectively, master problem), even if the time limit is exceeded in the meantime.

We set a time limit of 30 min for the master problem. If no feasible solution is obtained within this time, the algorithm execution continues until one is found. For the sub-problem, it is sufficient to limit the time to 3 min. Here, we take advantage of the fact that it can be solved individually for each scenario instead of considering them all at once. This significantly reduces the computational burden, but it also means that the solution time of the sub-problem scales with the number of scenarios, as they are

Table 3 Runtimes for the different formulations and instances in seconds, averaged over all uncertainty set sizes and values for Γ'

model	small	lowersaxony	human	mouse	gte
<i>MA</i>	382.56	6200.08	3687.06	78.04	11,289.26
<i>LA</i>	52,873.67	47,353.45	60,309.20	84,434.52	66,440.38
naive	788.61	1817.98	2481.08	831.86	3825.67

solved sequentially, although this could be overcome by parallelization. For the naive approach, we use a maximum of one hour per scenario.

As already indicated, the adjustable version of MWIFDP is computationally challenging, which is what motivates the employment of the column-and-constraint generation algorithm. Indeed, the master problem alone, consisting of an adjustable MWIFDP (with a reduced number of scenarios that increases as iterations progress), is in itself computationally demanding. While in *MA*, there is only one set of non-adjustable variables, and thus x and w can be chosen individually depending on the scenario, in *LA*, both y and x must be feasible (and in the best case optimal) for all scenarios. This leads to the less adjustable formulation *LA* being significantly harder to solve, which is reflected in the runtimes and number of iterations reported in Tables 3 and 4, respectively. The numbers provided refer to the average of all uncertainty set sizes $|\mathcal{U}(\Gamma)|$ and values for Γ' . Note that we use a pre-solving phase to initialize the value for the lower bound, which we do not count in the number of iterations. More precisely, after initializing $|\mathcal{U}(\Gamma)|$, we solve the master problem without upper bounds on the flow values and also without $\mathcal{W}$ in the objective function (corresponding to Lemma 13). We note that this pre-solving step is structurally equivalent to solving the strictly robust counterpart of the MWIFDP without upper bounds, as established in Sect. 5.2, and therefore, the computational study implicitly encompasses the strict robustness case. The resulting value for the number of paths is used to initialize *LB*.

While *MA* rarely reaches the time limit and usually requires only a few iterations to converge, the opposite is true for *LA*. The particularly low mean runtimes of *MA* for *small* and *mouse* can be explained by the small graph size and simple structure of the graph (very long paths without branches), respectively. For *LA*, however, these are dominated by the additional challenge incurred by the less adjustable master problem. Moreover, the fact that with *MA*, we have two adjustable variable sets causes the sub-problem to no longer be solved in a few seconds, as with *LA*, where only the optimal values for w need to be computed. However, since we can consider the sub-problem individually for each scenario, the computational requirements are still moderate. Nevertheless, the fact that we do not solve the scenarios in parallel means that the runtime for *MA* can be significantly longer for larger uncertainty set sizes. As a consequence, some of the solutions showcased below are not necessarily optimal.

Table 4 Iterations for the different formulations and instances, averaged over all uncertainty set sizes and values for Γ'

model	small	lowersaxony	human	mouse	gte
<i>MA</i>	2.00	4.72	1.72	1.17	2.94
<i>LA</i>	31.33	30.39	30.22	39.95	36.44

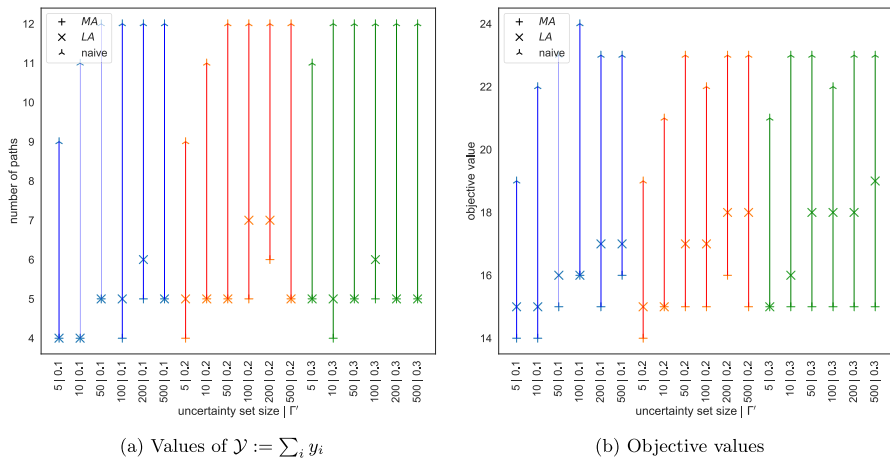


Fig. 4 Results of *MA*, *LA* and the naive approach for instance set *small* for the different uncertainty set sizes $|\mathcal{U}(\Gamma')|$ and values for Γ'

For the sake of clarity, we only show detailed results for the instance sets *small*, *lowsaxony*, and *human*. All other results can be found in Appendix 2. For *small*, Fig. 4 shows both the values for $\sum_i y_i := \mathcal{Y}$ (number of paths) and the objective values for the different uncertainty set sizes and values for Γ . The objective values for *lowsaxony* and *human* are shown in Fig. 5a and b, respectively. As we can see, the values for *MA* are consistently the lowest, and those of the naive approach are the highest. This also aligns with theoretical reasoning, which states that the less adjustable formulation *LA* can never achieve better objective values than *MA* and is bounded from above by the objective function value of the naive approach. These theoretical properties apply to the objective value but not necessarily to the number of paths. Nevertheless, the experimental results also show the same behavior for the latter.

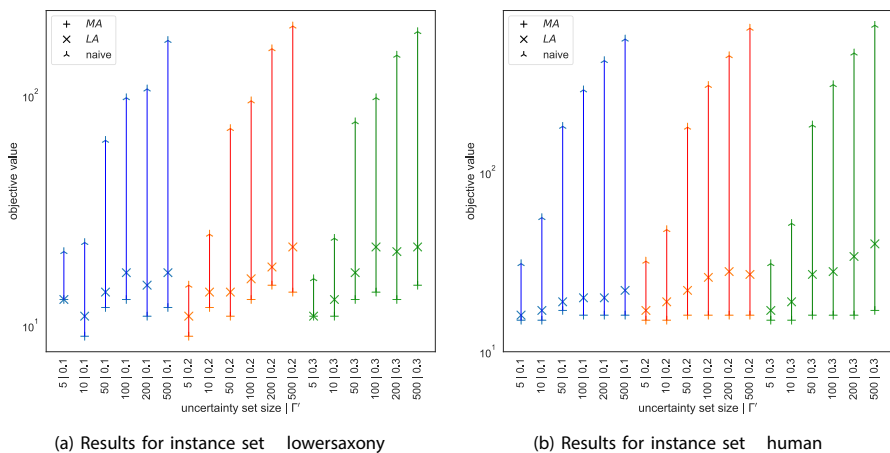


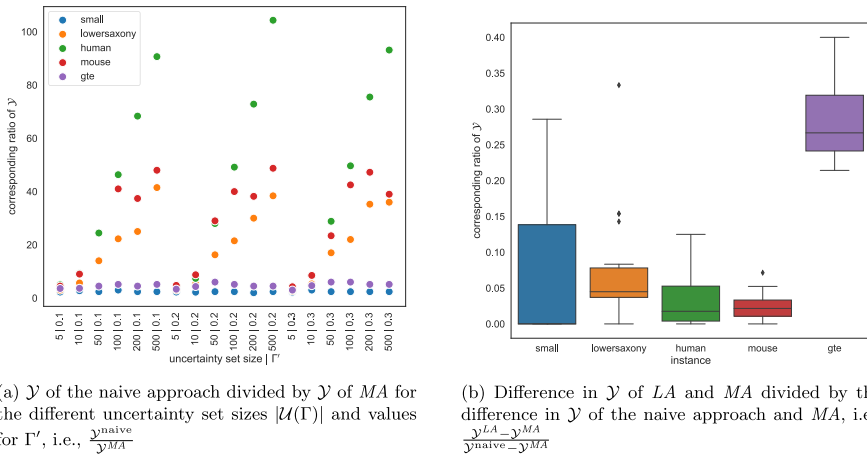
Fig. 5 Objective values of *MA*, *LA* and the naive approach for the different uncertainty set sizes $|\mathcal{U}(\Gamma')|$ and values for Γ' (logarithmic scale)

Since we use a set of s - t -paths of size $|\mathcal{P}'|$ to create a scenario, we know that we can decompose each scenario into a maximum of $|\mathcal{P}'|$ s - t -paths. This means that in MA , $\sum_i y_i$ is bounded by $|\mathcal{P}'|$ since we only have to determine the maximum number of paths that is sufficient per scenario. For LA , in addition to the number, the paths themselves must also be specified, forming a feasible solution for all scenarios. Consequently, the union of the original s - t -paths of all scenarios is feasible, i.e., $\sum_i y_i$ is bounded by $|\mathcal{U}(\Gamma)| \cdot |\mathcal{P}'|$. With this theoretical background, however, it can be seen from the results that these bounds are not even close to being reached. Depending on the instance, the maximum number of paths ranges between 5 and 8 for MA , 7 and 19 for LA , and 12 to 652 for the naive approach, while the theoretical upper bound is 5000.

In terms of the summed weights $\mathcal{W}$, the results of MA and the naive approach are similar (between 6 and 12) and hardly vary for different values of $|\mathcal{U}(\Gamma)|$ and Γ' . This can be explained by the fact that with both methods, the paths and weights can be adjusted to each other and chosen to suit the scenario. In contrast, LA searches for a set of paths that fits all scenarios. Consequently, it becomes harder to match the paths with the different values for w to the individual scenario, which leads to significantly higher values of $\mathcal{Y}$ and $\mathcal{W}$ (between 6 and 21) and also to both increasing with growing $|\mathcal{U}(\Gamma)|$ and Γ' .

When looking at $\mathcal{Y}$, we see a similar picture for MA as we did before with $\mathcal{W}$. Due to the more comprehensive adjustability, the values are relatively small (between 3 and 8), and the increase with growing $|\mathcal{U}(\Gamma)|$ and Γ' is also modest. The opposite is the case with the naive approach. Since the set of paths is generated by the union of the solutions for the individual scenarios, in order to be feasible for all scenarios (as in LA), $\mathcal{Y}$ increases with $|\mathcal{U}(\Gamma)|$ (up to a value of 652 and significantly more than in LA), which also leads to the objective value being dominated by $\mathcal{Y}$. This is particularly noticeable with more complex (but not necessarily larger) graphs, where the number of different paths (and thus the objective value) is significantly higher and can be well observed in Figs. 5 and 6a. In the latter, one can see how the ratio of the number of paths between the naive approach and MA increases significantly for the more complex graphs and with larger values for $|\mathcal{U}(\Gamma)|$ (depending on the instance, the naive approach requires up to 104 times more paths than MA). This underlines the advantage of our adjustable models, which provide a substantial improvement compared to the naive approach. Even the less adjustable version LA leads to a significant reduction in the number of paths, which can be seen in Fig. 6b. It shows the box plots for the five different instance classes. These associated values are calculated from the difference in the number of paths of LA and MA divided by the difference in the number of paths of the naive approach and MA . In other words, it indicates as a percentage how close the $\mathcal{Y}$ value of version LA is to the naive approach starting from MA . As can be seen in Fig. 6b, almost all values are below 0.15. The only exception is instance *gte*, for which the corresponding values range between 0.2 and 0.4, which is still a significant improvement compared to the naive approach.

To summarize the results, it can be noted that the adjustability has a considerable advantage over the non-adjustable model, and this is clearly reflected in the number of paths. Furthermore, even in the less adjustable LA case, our models deliver significantly better results than the naive approach at the expense of dealing with more computationally challenging problems.



(a) $\mathcal{Y}$ of the naive approach divided by $\mathcal{Y}$ of MA for the different uncertainty set sizes $|\mathcal{U}(\Gamma)|$ and values for Γ' , i.e., $\frac{\mathcal{Y}^{naive}}{\mathcal{Y}^{MA}}$

(b) Difference in $\mathcal{Y}$ of LA and MA divided by the difference in $\mathcal{Y}$ of the naive approach and MA , i.e., $\frac{\mathcal{Y}^{LA} - \mathcal{Y}^{MA}}{\mathcal{Y}^{naive} - \mathcal{Y}^{MA}}$

Fig. 6 Values of $\mathcal{Y} := \sum_i y_i$ (number of paths) of the three algorithms in relation to each other for the different instance sets

7 Conclusions

In this paper, we address the minimum flow decomposition problem and consider it from the perspective of robust optimization. Unlike typical bioinformatics applications that focus on robustness against measurement errors, we explore classical robustness concepts from the literature. Our focus also lies on applications related to transportation planning problems, such as optimizing vehicle utilization in public transport.

Motivated by these applications, we generalize the problem to the weighted inexact case with lower and upper bounds on the flow values. We investigate the complexity of the resulting problem and show that we can solve the special case where we only minimize the weights in polynomial time. Moreover, we introduce the concept of robust flow decomposition by incorporating uncertain bounds. Here, we show that arbitrary uncertainty sets can be reduced to finite, discrete uncertainty sets and that for strict robustness, the uncertainty set reduces to a single worst-case scenario.

To overcome the conservatism of strict robustness, and motivated by practical applications, we make use of adjustable robustness. Consequently, we present two different adjustable problem formulations. In the first, only the number of paths needs to be specified beforehand. In the second formulation, both the number and the specific paths must be decided before the scenario is revealed.

Our computational study provides a proof of concept, demonstrating the benefits of adjustability in settings under uncertainty when compared to non-adjustable models, which is clearly reflected in the number of paths. Even the less adjustable model yields significantly better results than the approach in which the problem is solved separately for each scenario, and all paths with corresponding weights are aggregated into a pool. Our findings highlight the substantial advantages of our adjustable models.

However, the gained flexibility comes at the cost of higher computational requirements. Thus, focusing on specialized solution methods that aim at improving runtimes would be a valuable contribution to future research. In particular, the column-and-

constraint generation approach employed shares structural similarities with Benders decomposition, and the rich literature on algorithmic enhancements to the latter, including combinatorial Benders cuts [13, 31, 37] and problem-specific decomposition strategies [32, 38], offers promising directions for improving the computational performance of the proposed framework.

Moreover, methods that can influence the choice of effective scenarios, both for initializing the algorithm and during its progression, based on external factors (such as data, as proposed in [19]) could play a significant role in improving computational performance. Moreover, exploring alternative uncertainty set formulations may expose not only better trade-offs in terms of robustness guarantees but also favor computational performance. Finally, since previous concepts have mainly focused on dealing with data inaccuracies, and this is the first work to consider MFDs in the context of classical robustness concepts, this opens up additional fields of application and offers opportunities for further research.

Appendix 1. Adjustable formulations

A.1 Master Problem and Sub-Problem for MA

Formulations of the master problem Eq. 22 and sub-problem Eq. 23 for MA in Algorithm 1 in Sect. 5.3.

$$\min_{x,y,w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (22a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \bar{\mathcal{U}} \quad (22b)$$

$$\sum_{v:(s,v) \in E} x_{svi}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \bar{\mathcal{U}} \quad (22c)$$

$$\sum_{u:(u,t) \in E} x_{uti}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \bar{\mathcal{U}} \quad (22d)$$

$$\sum_{(u,v) \in E} x_{uvi}^\xi - \sum_{(v,w) \in E} x_{vwi}^\xi = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\}, \xi \in \bar{\mathcal{U}} \quad (22e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \geq f_{uv}^{l,\xi}, \quad \forall (u, v) \in E, \xi \in \bar{\mathcal{U}} \quad (22f)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \leq f_{uv}^{u,\xi}, \quad \forall (u, v) \in E, \xi \in \bar{\mathcal{U}} \quad (22g)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (22h)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \bar{\mathcal{U}} \quad (22i)$$

$$x_{uvi}^\xi \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\}, \xi \in \bar{\mathcal{U}} \quad (22j)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (22k)$$

$$\min_{x,y,w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (23a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \mathcal{U} \quad (23b)$$

$$\sum_{v:(s,v) \in E} x_{svi}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (23c)$$

$$\sum_{u:(u,t) \in E} x_{uti}^\xi = y_i, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (23d)$$

$$\sum_{(u,v) \in E} x_{uvi}^\xi - \sum_{(v,w) \in E} x_{vwi}^\xi = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\}, \xi \in \mathcal{U} \quad (23e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \geq f_{uv}^{l,\xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (23f)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi}^\xi \leq f_{uv}^{u,\xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (23g)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (23h)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (23i)$$

$$x_{uvi}^\xi \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U}. \quad (23j)$$

A.2 Adjustable Formulation LA

Adjustable MIP formulation Eq. 24 for LA in Sect. 5.3, where the wait-and-see variables w^ξ depend on the scenario $\xi \in \mathcal{U}$ (given by the bounds $f^{l,\xi}$ and $f^{u,\xi}$).

$$\min_{x,y,w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (24a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \mathcal{U} \quad (24b)$$

$$\sum_{v:(s,v) \in E} x_{svi} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\} \quad (24c)$$

$$\sum_{u:(u,t) \in E} x_{uti} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\} \quad (24d)$$

$$\sum_{(u,v) \in E} x_{uvi} - \sum_{(v,w) \in E} x_{vwi} = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\} \quad (24e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \geq f_{uv}^{l,\xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (24f)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \leq f_{uv}^{u, \xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (24g)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (24h)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U} \quad (24i)$$

$$x_{uvi} \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\} \quad (24j)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (24k)$$

The corresponding column-and-constraint generation approach (based on [54]) to solve the adjustable problem LA in Eq. 24 can be found in Algorithm 3.

As described in Sect. 5.3, if a given solution (y^k, x^k) for the master problem leads to an infeasible sub-problem, we know that we can eliminate the combination of (y^k) and (x^k) from the solution space. As a consequence, in line 13 of Algorithm 3, we add the corresponding constraint to the master problem Eq. 25.

Algorithm 3 Column-and-constraint generation method for LA .

```

1: given a scenario set  $\mathcal{U}$  and threshold  $\epsilon$ 
2: Set  $LB = -\infty, UB = \infty, k = 1$  and initialize  $\bar{\mathcal{U}}$  with one scenario from  $\mathcal{U}$ 
3: while  $UB - LB > \epsilon$  do
4:   solve master problem Eq. 25, get optimal solution  $(y^k, x^k, w^k, \mathcal{W}^k)$ 
5:   update lower bound  $LB = \sum_{i=1}^{\bar{K}} (y^k)_i + \mathcal{W}^k$ 
6:   solve sub-problem Eq. 26 with input  $(y^k, x^k)$ 
7:   if we obtain an optimal solution  $(\bar{w}^k, \bar{\mathcal{W}}^k)$  then
8:     let  $\xi^k$  be the worst-case scenario causing  $\bar{\mathcal{W}}^k$ 
9:     update upper bound  $UB = \sum_{i=1}^{\bar{K}} (y^k)_i + \bar{\mathcal{W}}^k$ 
10:  else
11:    let  $\xi^k$  be the first scenario causing infeasibility
12:    add  $\sum_{i: (y^k)_i=0} y_i + \sum_{i: (x^k)_i=0} x_i + \sum_{i: (y^k)_i=1} (1 - y_i) + \sum_{i: (x^k)_i=1} (1 - x_i) \geq 1$  to constraints
    of master problem Eq. 25
13:    add  $\xi^k$  to  $\bar{\mathcal{U}}$ 
14:  end if
15:   $k = k + 1$ 
16: end while
17: return  $(y^k, x^k, \bar{w}^k)$ 

```

The described master problem and sub-problem of LA in Algorithm 3 are shown in Eqs. 25 and 26, respectively.

$$\min_{x, y, w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (25a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \bar{\mathcal{U}} \quad (25b)$$

$$\sum_{v: (s, v) \in E} x_{svi} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\} \quad (25c)$$

$$\sum_{u:(u,t) \in E} x_{uti} = y_i, \quad \forall i \in \{1, \dots, \bar{K}\} \quad (25d)$$

$$\sum_{(u,v) \in E} x_{uvi} - \sum_{(v,w) \in E} x_{vwi} = 0, \quad \forall i \in \{1, \dots, \bar{K}\}, v \in V \setminus \{s, t\} \quad (25e)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \geq f_{uv}^{l, \xi}, \quad \forall (u, v) \in E, \xi \in \bar{\mathcal{U}} \quad (25f)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \leq f_{uv}^{u, \xi}, \quad \forall (u, v) \in E, \xi \in \bar{\mathcal{U}} \quad (25g)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (25h)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \bar{\mathcal{U}} \quad (25i)$$

$$x_{uvi} \in \{0, 1\}, \quad \forall (u, v) \in E, i \in \{1, \dots, \bar{K}\} \quad (25j)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, \bar{K}\}. \quad (25k)$$

$$\min_{x, y, w} \sum_{i=1}^{\bar{K}} y_i + \mathcal{W} \quad (26a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\bar{K}} w_i^\xi \leq \mathcal{W}, \quad \forall \xi \in \mathcal{U} \quad (26b)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \geq f_{uv}^{l, \xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (26c)$$

$$\sum_{i \in \{1, \dots, \bar{K}\}} w_i^\xi x_{uvi} \leq f_{uv}^{u, \xi}, \quad \forall (u, v) \in E, \xi \in \mathcal{U} \quad (26d)$$

$$\mathcal{W} \in \mathbb{Z}^+ \quad (26e)$$

$$w_i^\xi \in \mathbb{Z}^+, \quad \forall i \in \{1, \dots, \bar{K}\}, \xi \in \mathcal{U}. \quad (26f)$$

Appendix 2. Detailed results

Complementing Sect. 6.2, the remaining results can be found in the following. For *mouse* and *gte*, Figs. 7 and 8 show both the values for $\mathcal{Y} := \sum_i y_i$ and the objective values for the different uncertainty set sizes and values for Γ .

The values for $\mathcal{Y}$ for *lowersaxony* and *human* are shown in Fig. 9a and b, respectively.

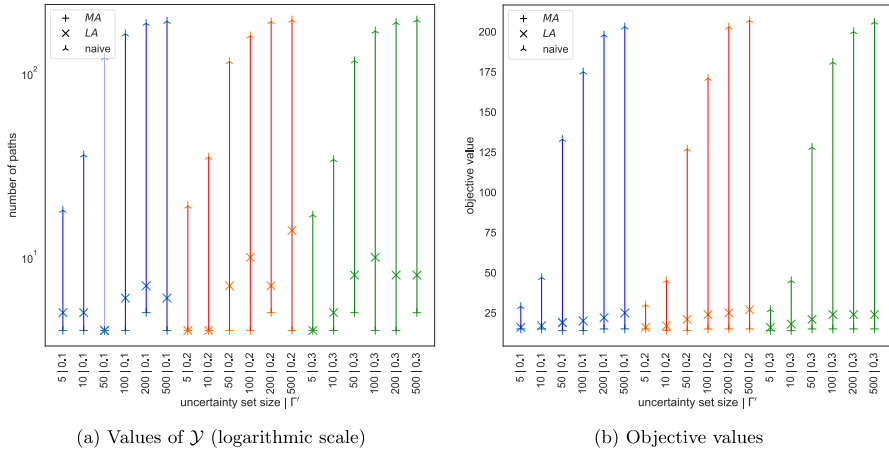


Fig. 7 Results of *MA*, *LA* and the naive approach for instance set *mouse* for the different uncertainty set sizes $|\mathcal{U}(\Gamma)|$ and values for Γ'

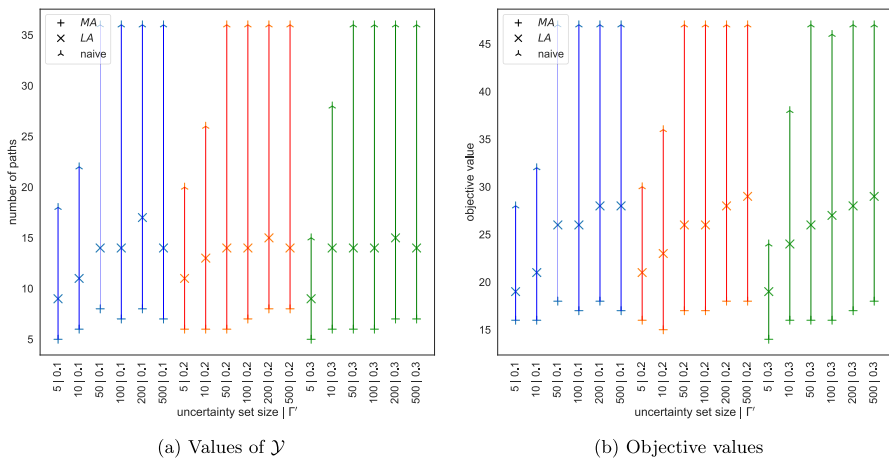


Fig. 8 Results of *MA*, *LA* and the naive approach for instance set *gte* for the different uncertainty set sizes $|\mathcal{U}(\Gamma)|$ and values for Γ'

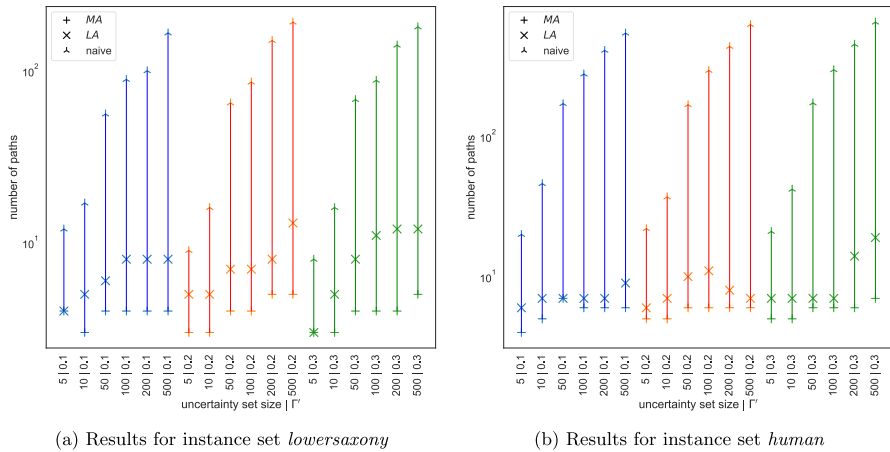


Fig. 9 Values for $\mathcal{Y}$ of MA, LA and the naive approach for the different uncertainty set sizes $|\mathcal{U}(\Gamma)|$ and values for Γ' (logarithmic scale)

Author Contribution M.S. is the main author, taking responsibility for writing the manuscript, preparing all visual materials, developing the technical analyses, and executing the computational experiments. P.S. contributed by revising the manuscript, providing essential technical support for the analysis, and contributing to delimiting the scope and designing the computational studies. F.O. contributed by revising the manuscript, delimiting the scope, and designing the computational studies.

Funding Open access funding provided by Technical University of Denmark This work was supported by the Research Council of Finland (Flagship of Advanced Mathematics for Sensing Imaging and Modelling 359181).

Data Availability The code and data generated are available at www.github.com/gamma-opt/robust-flow-decomposition.

Declarations

Ethics Approval Not applicable

Consent to Participate Not applicable

Consent for Publication Not applicable

Conflict of Interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ahuja RK, Magnanti TL, Orlin JB (1988) Network flows
- Arsilan O, Laporte G (2024) Network design with vulnerability constraints and probabilistic edge reliability. *Networks* 84(2):181–199. <https://doi.org/10.1002/net.22222>
- Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Vol 28. Princeton university press
- Ben-Tal A, Nemirovski A (1998) Robust convex optimization. *Math Oper Res* 23(4):769–805
- Ben-Tal A et al (2004) Adjustable robust solutions of uncertain linear programs. *Math Program* 99(2):351–376
- Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev* 53(3):464–501
- Bertsimas D, Sim M (2004) The price of robustness. *Oper Res* 52(1):35–53
- Beyer H-G, Sendhoff B (2007) Robust optimization-a comprehensive survey. *Comput Methods Appl Mech Eng* 196(33–34):3190–3218
- Bunte S, Kliever N (2009) An overview on vehicle scheduling models. *Public Transport* 1(4):299–317
- Cacchiani V, Qi J, Yang L (2020) Robust optimization models for integrated train stop planning and timetabling with passenger demand uncertainty. *Trans Res Part B Methodol* 136:1–29
- Cacchiani V, Toth P (2012) Nominal and robust train timetabling problems. *Eur J Oper Res* 219(3):727–737
- Casazza M, Ceselli A (2021) Optimization algorithms for resilient path selection in networks. *Comput Oper Res* 128:105191. <https://doi.org/10.1016/j.cor.2020.105191>
- Codato G, Fischetti M (2006) Combinatorial benders' cuts for mixed-integer linear programming. *Oper Res* 54(4):756–766. <https://doi.org/10.1287/opre.1060.0286>
- Dias FHC, Tomescu AI (2023) Accurate flow decomposition via robust integer linear programming. *bioRxiv*, pp 2023–03
- Dias FHC et al (2022) Efficient minimum flow decomposition via integer linear programming. *J Comput Biol* 29(11):1252–1267
- Diestel R (2005) Graph theory 3rd ed. *Grad Texts Math* 173(33):12
- Gabrel V, Murat C, Thiele A (2014) Recent advances in robust optimization: an overview. *Eur J Oper Res* 235(3):471–483
- Michael RG, David SJ (1979) Computers and intractability: a guide to the theory of NP-completeness
- Goerigk M, Kurtz J (2023) Data-driven robust optimization using deep neural networks. *Comput Oper Res* 151:106087
- Goldberg AV, Tardos E, Tarjan R (1989) Network flow algorithm. Tech. rep. Cornell University Operations Research and Industrial Engineering
- Guihaire V, Hao J-K (2008) Transit network design and scheduling: A global review. *Trans Res Part A Pol Pract* 42(10):1251–1273
- Hartman T et al (2012) How to split a flow? In: 2012 Proceedings IEEE INFOCOM. IEEE, pp 828–836
- Huisman D et al (2005) Operations research in passenger railway transportation. *Stat Neerl* 59(4):467–497
- Kouvelis P, Yu G (2013) Robust discrete optimization and its applications. Vol 14. Springer Science & Business Media
- Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *J Oper Res Soc* 63(9):1294–1306
- Ley E et al (2023) Robust optimization under controllable uncertainty
- Li B et al (2013) A survey of network flow applications. *J Netw Comput Appl* 36(2):567–581
- Li W, Feng J, Jiang T (2011) IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *J Comput Biol* 18(11):1693–1707
- Lusby RM, Larsen J, Bull S (2018) A survey on robustness in railway planning. *Eur J Oper Res* 266(1):1–15
- Lusby RM et al (2011) Railway track allocation: models and methods. *OR Spectrum* 33:843–883
- Mao Z et al (2024) Combinatorial Benders decomposition for single machine scheduling in additive manufacturing with two-dimensional packing constraints. *Eur J Oper Res* 317(3):890–905. <https://doi.org/10.1016/j.ejor.2024.05.001>
- Michels AS et al (2019) A Benders' decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem. *Eur J Oper Res* 278(3):796–808. <https://doi.org/10.1016/j.ejor.2019.05.001>, <https://www.sciencedirect.com/science/article/pii/S0377221719303844>

33. Ntafos SC, Hakimi SL (1979) On path cover problems in digraphs and applications to program testing. *IEEE Trans Software Eng* 5:520–529
34. Olsen N, Klierer N, Wolbeck L (2022) A study on flow decomposition methods for scheduling of electric buses in public transport based on aggregated time–space network models. *Central Eur J Oper Res* 1–37
35. Orlin J (1988) A faster strongly polynomial minimum cost flow algorithm. In: *Proceedings of the Twentieth annual ACM symposium on theory of computing*, pp 377–387
36. Parbo J, Nielsen OA, Prato CG (2016) Passenger perspectives in railway timetabling: a literature review. *Transp Rev* 36(4):500–526
37. Rahmaniani R et al (2017) The Benders decomposition algorithm: a literature review. *Eur J Oper Res* 259(3):801–817. <https://doi.org/10.1016/j.ejor.2016.12.005>
38. Rist Y, Forbes M (2022) A column generation and combinatorial benders decomposition algorithm for the selective dial-a-ride-problem. *Comput Oper Res* 140:105649. <https://doi.org/10.1016/j.cor.2021.105649>, <https://www.sciencedirect.com/science/article/pii/S0305054821003531>
39. Salimifard K, Bigharaz S (2022) The multicommodity network flow problem: state of the art classification, applications, and solution methods. *Oper Res Int J* 22(1):1–47
40. Schiewe P, Stinzendörfer M (2022) Integrated line planning and vehicle scheduling for public transport. In: *INOC*, pp 1–6
41. Schiewe P et al (2024) LinTim: An integrated environment for mathematical public transport optimization. Course material. <https://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-83557>
42. Schöbel A (2014) Generalized light robustness and the trade-off between robustness and nominal quality. *Math Methods Oper Res* 80(2):161–191
43. Schöbel A (2012) Line planning in public transportation: models and methods. *OR Spectrum* 34(3):491–510
44. Mingfu Shao and Carl Kingsford (2017) Theory and a heuristic for the minimum path flow decomposition problem. *IEEE/ACM Trans Comput Biol Bioinf* 16(2):658–670
45. Soyster AL (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper Res* 21(5):1154–1157
46. Tomescu AI et al (2015) Explaining a weighted DAG with few paths for solving genome-guided multi-assembly. *IEEE/ACM Trans Comput Biol Bioinf* 12(6):1345–1354
47. Vatinlen B et al (2008) Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. *Eur J Oper Res* 185(3):1390–1401
48. Veldhorst M (1993) A bibliography on network flow problems. In: *Network optimization problems: algorithms, applications and complexity*. World Scientific, pp 301–331
49. Wayne KD (1999) A polynomial combinatorial algorithm for generalized minimum cost flow. In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pp 11–18
50. Wei H-Z, Chen C-R, Li S-J (2018) Characterizations for optimality conditions of general robust optimization problems. *J Optim Theory Appl* 177:835–856
51. Williams L, Reynolds G, Mumey B (2019) RNA transcript assembly using inexact flows. In: *2019 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, pp 1907–1914
52. Xing Y, Resch A, Lee C (2004) The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures. *Genome Res* 14(3):426–441
53. Yamkoğlu İ, Gorissen BL, den Hertog D (2019) A survey of adjustable robust optimization. *Eur J Oper Res* 277(3):799–813
54. Zeng B, Zhao L (2013) Solving two-stage robust optimization problems using a column-and-constraint generation method. *Oper Res Lett* 41(5):457–461
55. Zhu X et al (2011) Minimal-cost network flow problems with variable lower bounds on arc flows. *Comput Oper Res* 38(8):1210–1218