Annals of Operations Research (2022) 312:691–722
https://doi.org/10.1007/s10479-021-04426-0

ORIGINAL RESEARCH

Robust minimum cost flow problem under consistent flow constraints

Christina Büsing[1] · **Arie M. C. A. Koster**[1] · **Sabrina Schmitz**[1]

Accepted: 10 November 2021 / Published online: 28 December 2021

Abstract
The robust minimum cost flow problem under consistent flow constraints (RobMCF≡) is a new extension of the minimum cost flow (MCF) problem. In the RobMCF≡ problem, we consider demand and supply that are subject to uncertainty. For all demand realizations, however, we require that the flow value on an arc needs to be equal if it is included in the predetermined arc set given. The objective is to find feasible flows that satisfy the equal flow requirements while minimizing the maximum occurring cost among all demand realizations. In the case of a finite discrete set of scenarios, we derive structural results which point out the differences with the polynomial time solvable MCF problem in networks with integral demands, supplies, and capacities. In particular, the Integral Flow Theorem of Dantzig and Fulkerson does not hold. For this reason, we require integral flows in the entire paper. We show that the RobMCF≡ problem is strongly $\mathcal{NP}$-hard on acyclic digraphs by a reduction from the $(3, B2)$-SAT problem. Further, we demonstrate that the RobMCF≡ problem is weakly $\mathcal{NP}$-hard on series-parallel digraphs by providing a reduction from PARTITION. If in addition the number of scenarios is constant, we propose a pseudo-polynomial algorithm based on dynamic programming. Finally, we present a special case on series-parallel digraphs for which we can solve the RobMCF≡ problem in polynomial time.

Keywords Minimum cost flow problem · Equal flow problem · Robust flows · Series-parallel digraphs · Dynamic programming

1 Introduction

In this paper, we present a new extension of the minimum cost flow (MCF) problem (Ahuja et al. 1988), which we call the *robust minimum cost flow problem under consistent flow constraints* (RobMCF≡). This problem is motivated by, for example, long-term decisions

✉ Sabrina Schmitz
schmitz@math2.rwth-aachen.de

Christina Büsing
buesing@math2.rwth-aachen.de

Arie M. C. A. Koster
koster@math2.rwth-aachen.de

[1] Lehrstuhl II für Mathematik, RWTH Aachen University, Pontdriesch 10-12, 52062 Aachen, Germany

in logistic applications. A major problem in logistics is the cost-efficient transport of commodities. Typically, this problem is represented by an MCF model, where a commodity can be identified by a flow sent through a network from a supply source to a sink with demand. In this way, a company can easily assess whether the available means of transport are sufficient for a given demand. If this is not the case, additional transport by subcontractors can be arranged. Such arrangements are generally agreed by long-term contracts, however, the demand is naturally subject to uncertainty. For this reason, valid and cost-efficient decisions have to be made without the knowledge of the actual demand.

The problem described can be represented by an adjusted integral MCF model subject to demand uncertainty. We represent the demand uncertainty by a discrete number of possible occurring demand scenarios. In addition to the network requirements of the MCF problem, we are given a predetermined set of arcs referred to as fixed arcs. The flow value on the fixed arcs is supposed to represent the transport by subcontractors. Thus, we require the flow value on a fixed arc to be equal among all demand scenarios. For finding a robust solution to this problem, we minimize the maximum cost that may occur among all demand scenarios. In summary, for the RobMCF$\equiv$ problem, we consider different demand scenarios for which we require integral flows whose flow values are equal on the respective fixed arcs with the objective of minimizing the maximum cost.

The main contribution of this paper can be summarized as follows. We show that most of the knowledge of the MCF problem is not readily transferable to the RobMCF$\equiv$ problem. In particular, the integrality requirement of the RobMCF$\equiv$ problem is necessary, even though the demands and supplies of all demand scenarios and the network's capacities are integral, as Dantzig and Fulkerson's Integral Flow Theorem (Korte et al. 2012) does not hold. We further prove that finding a feasible solution to the RobMCF$\equiv$ problem is strongly $\mathcal{NP}$-complete on acyclic digraphs, even if only two demand scenarios and unit arc capacities are considered. On series-parallel digraphs, we show that the decision version of the RobMCF$\equiv$ problem is weakly $\mathcal{NP}$-complete and in the special case of a constant number of scenarios it is solvable in pseudo-polynomial time by dynamic programming. If all demand scenarios have the same single source and sink, we propose an algorithm running in polynomial time on series-parallel digraphs.

The outline of this paper is as follows. We start with an overview about related work in Sect. 2. Subsequently, in Sect. 3, we give an explicit mathematical problem description, and introduce the notations of this paper. Furthermore, we present first structural results of the problem. In Sect. 4, the problem's complexity is analyzed on acyclic digraphs. Afterwards, we consider the RobMCF$\equiv$ problem on series-parallel digraphs in Sect. 5. We conclude this paper by Sect. 6.

2 Related work

In the context of logistic applications, several extensions to the MCF problem are analyzed in the literature. We refer to the *minimum cost flows with minimum quantities problem* introduced by Seedig (2011) and generalized by Krumke and Thielen (2011). In addition to the requirements of the MCF problem, on each arc the flow must either take the value zero or a minimum quantity given. This can be altered and transfered to the case where the flow on the fixed arcs, which represents the transport by subcontractors, takes a minimum value (among all demand scenarios). Another extension is obtained, for example, by the MCF version of the *maximum flow problem with disjunctive constraints* studied by Pferschy and Schauer (2013).

A maximum flow is sought whose flow value is positive for at least one arc of every arc pair included in a predetermined arc set. Thus, at least one transport performed by subcontractors can be modeled. We point out that the problems above are only considered for one demand scenario, whereas we consider several demand scenarios in the RobMCF≡ problem. Due to the equal flow requirements for the fixed arcs, we focus in the following on extensions in the literature with equal flow requirements. Afterwards, we give a short overview of robust network approaches with demand uncertainty. To the best of our knowledge, no study combines equal flow requirements with robust network approaches.

Sahni (1974) introduces a variant of the maximum flow (MF) problem (Ahuja et al. 1988), the so-called *integral flow with homologous arcs problem* (HOMIF). In addition to the set-up of the MF problem, predetermined sets of arcs are given in this problem. A maximum integral flow is sought whose flow value is equal on all arcs that are contained in the same predetermined arc set. Sahni proves the $\mathcal{NP}$-hardness of the problem by a reduction from the NON- TAUTOLOGY problem. Johnson and Garey (1979) point out that by modifying a construction of Even et al. (1975), the problem's $\mathcal{NP}$-hardness holds even if all arc capacities are equal to one. Furthermore, unless $\mathcal{P} = \mathcal{NP}$, the non-existence of a $2^{n(1-\varepsilon)}$-approximation algorithm for any fixed $\varepsilon > 0$ (on a digraph with n vertices) is proven by Meyers and Schulz (2009) even if a nonzero solution is guaranteed to exist.

The MCF version of the HOMIF problem can be found in the literature as (integer) *equal flow problem* (EF). Using standard techniques, the complexity results can be transformed from the MF to the MCF version (Ahuja et al. 1988). There are several special cases and applications for both the MF and MCF version of the problem considered in the literature (Calvete 2003; Meyers and Schulz 2009; Morrison et al. 2013; Srinathan et al. 2002). For instance, the special case of the EF problem where all sets have cardinality two, i.e., an integral MCF is sought whose flow value is equal on a predetermined set of arc pairs, is investigated by Ali et al. (1988). The problem finds application in, for example, crew scheduling (Carraresi and Gallo 1984). Therefore, Ali et al. present a heuristic algorithm based on Lagrangian relaxation. Meyers and Schulz (2009) refer to this problem as *paired integer equal flow problem* (PEF) and prove that there exists no $2^{n(1-\varepsilon)}$-approximation algorithm for any fixed $\varepsilon > 0$ (on a digraph with n vertices), unless $\mathcal{P} = \mathcal{NP}$. The statement holds true even if a nontrivial solution is guaranteed to exist. A simpler and in polynomial time solvable special case of the EF problem is the so-called *simple equal flow problem* (SEF), which is introduced by Ahuja et al. (1999). The SEF problem requires the same but not necessarily integral flow value on only a single predetermined set of arcs. The problem is motivated by the management of water resource systems (Manca et al. 2010). For this purpose, Ahuja et al. (1999) develop several efficient algorithms to solve large-scale instances—a network simplex, a parametric simplex, a combinatorial parametric, a binary search, and a capacity scaling algorithm. These algorithms can easily be modified to obtain integral solutions.

Unlike the previous research on MF and MCF problems with equal flow requirements, we consider in the RobMCF≡ problem not one demand scenario only, but several demand scenarios. For each of these scenarios, a feasible flow is sought. Furthermore, among all of these scenarios, we require the same flow value on an arc if it is included in the single predetermined set of fixed arcs. Although we consider several demand scenarios, the problem of finding a feasible solution to the RobMCF≡ problem can be modeled as a special case of the EF and PEF problem by means of graph copies. We point out that the equal flow requirements in the RobMCF≡ problem are only of importance while considering different demand scenarios, i.e., the flow value of a fixed arc has to be equal among all scenarios. In turn, the flow value of two fixed arcs may differ in one scenario. For this reason, the problem of finding a feasible solution to the RobMCF≡ problem cannot be modeled as the sEF problem, except for the special case where the predetermined arc set contains only one arc. Moreover, due to different objectives, the correspondence from the RobMCF≡ problem to the EF, sEF, and pEF problem only holds for finding a feasible solution.

Demand uncertainty is studied more frequently in the context of network design and network engineering in telecommunication or road networks for example. In robust network design, we have to decide on the capacities such that in all considered scenarios, the entire demand can simultaneously be routed. The cost of installing the capacities is supposed to be minimized.

In the single commodity case which was first studied by Minoux (1989) and by Sanità (2009), the flow between supply and demand vertices may differ among the scenarios as long as the capacities are satisfied. For discrete scenarios, a cut-based integer linear program formulation with a separation algorithm is proposed by Álvarez-Miranda et al. (2012). Cacchiani et al. (2016) present a branch-and-cut algorithm for two types of uncertainty sets, a discrete set of scenarios, and a polytope. Atamtürk and Zhang (2007) present a two-stage robust optimization approach where some capacity decisions have to be made before, and other after the demand realization. The decisions have to guarantee that in any case the demand can be routed through the network.

In the multi-commodity case, several studies propose different models and uncertainty sets. For instance, Altin et al. (2007, 2011) propose the so-called Hose uncertainty model, Belotti et al. (2008) in the context of statistical multiplexing, and Koster et al. (2013) the budget uncertainty set introduced by Bertsimas and Sim (2003, 2004). In these studies, the flow is sent proportionally with the demand. In case the flow can be adapted to the demand, a two-stage robust approach is followed. While Mattia (2013) studies dynamic routing, Poss and Raack (2013) suggest to use affine recourse options.

3 Robust minimum cost flow problem under consistent flow constraints

3.1 Definition and notation

The RobMCF≡ problem is an extension of the MCF problem where supply and demand are subject to uncertainty. The uncertainty is represented by a finite set of discrete scenarios Λ where we do not have any knowledge which scenario is realized. Considering these scenarios, we are given a *digraph* $G = (V, A)$ with vertex set V and arc set A. The set of arcs A is defined by two disjoint sets, i.e., $A = A^{\text{fix}} \cup A^{\text{free}}$, where we refer to arcs of set A^{fix} as *fixed arcs* and arcs of set A^{free} as *free arcs*. We use the notation $V(G)$, $A(G)$, $A^{\text{fix}}(G)$, and $A^{\text{free}}(G)$ to indicate the sets of vertices, arcs, fixed arcs, and free arcs of digraph G, respectively. Independent of the scenarios arc *capacities* $u : A(G) \to \mathbb{Z}_{\geq 0}$ and arc *cost* $c : A(G) \to \mathbb{Z}_{\geq 0}$ are given. In contrast, vertex *balances* $b^{\lambda} : V(G) \to \mathbb{Z}$ with $\sum_{v \in V} b^{\lambda}(v) = 0$ that define the supply and demand realizations are given for every scenario $\lambda \in \Lambda$, denoted by $\boldsymbol{b} = (b^1, \dots, b^{|\Lambda|})$. A positive balance indicates a *source* while a negative balance indicates a *sink*. Note that, in general, the source (sink) vertices do not necessarily have to be the same in every scenario. In case that each scenario has only one vertex with a positive (negative) balance, we refer to these sources as *single sources (sinks)*. If the single sources (sinks) are defined by the same vertex for every scenario, we say that the problem has a *unique source (sink)*. Combined, we obtain the *network* $(G, u, c, \boldsymbol{b})$.

Considering only a single scenario $\lambda \in \Lambda$, analogues to the MCF problem a b^λ-flow is defined by a function $f^\lambda : A(G) \rightarrow \mathbb{Z}_{\geq 0}$ that satisfies the *capacity constraints*

$$0 \leq f^\lambda(a) \leq u(a)$$

on all arcs $a \in A$ and the *flow balance constraints*

$$\sum_{a=(v,w)\in A} f^\lambda(a) - \sum_{a=(w,v)\in A} f^\lambda(a) = b^\lambda(v)$$

at every vertex $v \in V$. The cost of a b^λ-flow f^λ is defined by

$$c(f^\lambda) = \sum_{a\in A} c(a) \cdot f^\lambda(a).$$

To consider a set of scenarios Λ, we need to introduce a new definition of a flow, a so called robust $\boldsymbol{b}$-flow.

Definition 1 (Robust Flow) Given a network $(G = (V, A = A^{\text{fix}} \cup A^{\text{free}}), u, c, \boldsymbol{b})$, a robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, \dots, f^{|\Lambda|})$ is defined by a $|\Lambda|$-tuple of b^λ-flows $f^\lambda : A(G) \rightarrow \mathbb{Z}_{\geq 0}$ that satisfy the consistent flow constraints $f^\lambda(a) = f^{\lambda'}(a)$ on all fixed arcs $a \in A^{\text{fix}}$ for all scenarios $\lambda, \lambda' \in \Lambda$. The cost of a robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ is defined by the maximum flow cost among all scenarios, i.e., $c(\boldsymbol{f}) = \max_{\lambda\in\Lambda} c(f^\lambda)$.

We refer to the flow value on an arc of set A^{fix} as its *load*. Accordingly, the consistent flow constraints are satisfied if the load of a fixed arc is equal in every scenario. The RobMCF≡ problem can finally be formulated as follows.

Definition 2 (RobMCF≡) Given a network $(G = (V, A = A^{\text{fix}} \cup A^{\text{free}}), u, c, \boldsymbol{b})$, the *robust minimum cost flow problem under consistent flow constraints* (RobMCF≡) aims a robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, \dots, f^{|\Lambda|})$ of minimum cost.

Note that in case of a single scenario, i.e., $|\Lambda| = 1$, the RobMCF≡ problem corresponds to the MCF problem. Otherwise, however, there are major differences as the following section shows.

3.2 Structural results

In this section, we present structural results of the RobMCF≡ problem. In particular, differences to the MCF problem are pointed out where the main difference is the following. Given a network with integral capacities and balances, by Dantzig and Fulkerson (Korte et al. 2012) there always exists an optimal integral flow for the MCF problem. This useful integral flow property is assumed to be given in most studies. However, the integral flow property does not hold for the RobMCF≡ problem as the following example shows.

Example 1 For a set of two scenarios $\Lambda = \{1, 2\}$, let a network $(G, u, c, \boldsymbol{b})$ with capacity $u \equiv 1$ be given, where digraph G, its cost c, and the non-zero balances $\boldsymbol{b}$ are visualized in Fig. 1a. An optimal integral robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ is defined by a first scenario flow f^1 that sends one unit along path $v_1v_3v_5v_8$, and a second scenario flow f^2 that sends one unit each along path $v_1v_3v_5v_6$ and $v_1v_4v_5v_7v_6$. This results in cost of $c(\boldsymbol{f}) = 4$ as $c(f^1) = 4$ and $c(f^2) = 2 + 0 = 2$ hold true.

However, by neglecting the integral flow requirement, there exists a robust $\boldsymbol{b}$-flow $\tilde{\boldsymbol{f}} = (\tilde{f}^1, \tilde{f}^2)$ with cost of $c(\tilde{\boldsymbol{f}}) = 3$. Flow $\tilde{f}^1$ sends a half unit each along paths $v_1v_3v_5v_8$

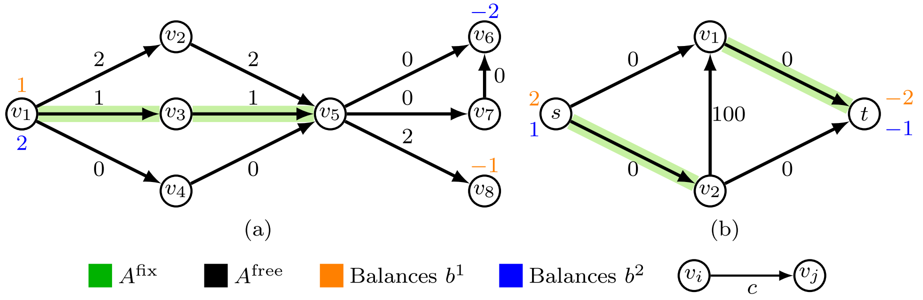

Fig. 1 **a** A non-integral robust ***b***-flow is the only optimal solution **b** A scenario flow that does not send the maximum demand among all scenarios causes the maximum cost in a unique source, unique sink network

and $v_1v_4v_5v_8$ ending up in total cost of $c(\tilde{f}^1) = 3$. Flow $\tilde{f}^2$ sends a half unit each along paths $v_1v_2v_5v_6$ and $v_1v_3v_5v_6$, and one unit along path $v_1v_4v_5v_7v_6$, also ending up in cost of $c(\tilde{f}^2) = 3$.

Corollary 1 *Considering the continuous relaxation of the RobMCF≡ problem, there does not always exist an integral robust flow with minimum cost, even if all capacities and balances are integral.*

We note that if no integer requirements for a robust flow are given, the RobMCF≡ problem can be solved by a simple linear program (LP) in polynomial time in $|V|$,$|A|$ and $|\Lambda|$. However, as applications of the RobMCF≡ problem often require integral flow values, hereafter this paper only concentrates on integral solutions. Further motivated by applications, in the next step, we investigate the RobMCF≡ problem where either the load of the fixed arcs is given, or the number of fixed arcs is constant. In logistics for example, this complies with finding a solution of minimum cost if the transport is already contractually agreed or limited. The following results show that we can solve these special cases in polynomial time.

Lemma 1 *Let $\mathcal{I} = (G = (V, A = A^{\text{fix}} \cup A^{\text{free}}), u, c, \boldsymbol{b})$ be a RobMCF≡ instance. For a given load $\ell : A^{\text{fix}}(G) \to \mathbb{Z}_{\geq 0}$, an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ that satisfies $f^{\lambda}(a) = \ell(a)$ for all fixed arcs $a \in A^{\text{fix}}(G)$ in every scenario $\lambda \in \Lambda$ can be computed in polynomial time if one exists.*

Proof We transform instance $\mathcal{I}$ to $|\Lambda|$ simple minimum cost $\tilde{b}^{\lambda}$-flow instances $\widetilde{\mathcal{I}}^{\lambda} = (\widetilde{G}, u, c, \tilde{b}^{\lambda})$ that can be considered for every scenario $\lambda \in \Lambda$ separately. Instances $\widetilde{\mathcal{I}}^{\lambda}$, $\lambda \in \Lambda$ are obtained by deleting the fixed arcs from digraph G resulting in digraph $\widetilde{G}$, i.e., $\widetilde{G} = G - A^{\text{fix}}$, while at the same time the new balances $\tilde{b}^{\lambda} : V(\widetilde{G}) \to \mathbb{Z}$ are defined as follows

$$\tilde{b}^{\lambda}(v) = b^{\lambda}(v) + \sum_{a=(w,v)\in A^{\text{fix}}(G)} \ell(a) - \sum_{a=(v,w)\in A^{\text{fix}}(G)} \ell(a).$$

After computing minimum cost $\tilde{b}^{\lambda}$-flows $\tilde{f}^{\lambda}$ for all instances $\widetilde{\mathcal{I}}^{\lambda}$, $\lambda \in \Lambda$, a corresponding robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, \ldots, f^{|\Lambda|})$ for instance $\mathcal{I}$ is defined as

$$f^{\lambda}(a) = \begin{cases} \ell(a) & \text{for all fixed arcs } a \in A^{\text{fix}}(G), \\ \tilde{f}^{\lambda}(a) & \text{for all free arcs } a \in A^{\text{free}}(G), \end{cases}$$

and causes cost of

$$c(\boldsymbol{f}) = \max_{\lambda \in \Lambda} c(\tilde{f}^{\lambda}) + \sum_{a \in A^{\text{fix}}(G)} c(a) \cdot \ell(a).$$

Assume the constructed robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ is not optimal, i.e., a robust $\boldsymbol{b}$-flow $\hat{\boldsymbol{f}} = (\hat{f}^1, \dots, \hat{f}^{|\Lambda|})$ exists with cost

$$c(\hat{f}^{\lambda_1}) = \max_{\lambda \in \Lambda} c(\hat{f}^{\lambda}) = c(\hat{\boldsymbol{f}}) < c(\boldsymbol{f}) = \max_{\lambda \in \Lambda} c(f^{\lambda}) = c(f^{\lambda_2})$$

for scenarios $\lambda_1, \lambda_2 \in \Lambda$. Let $\overline{f}^{\lambda}$ denote the flow which results from restricting the scenario flow $\hat{f}^{\lambda}$ of instance $\mathcal{I}$ to instance $\widetilde{\mathcal{I}}^{\lambda}$, $\lambda \in \Lambda$. As the load on the fixed arcs is given, the values of flows $\hat{\boldsymbol{f}}$ and $\boldsymbol{f}$ on the fixed arcs are equal for every scenario, i.e., $\hat{f}^{\lambda}(a) = f^{\lambda}(a) = \ell(a)$ for $a \in A^{\text{fix}}$ and $\lambda \in \Lambda$. Using this insight, we obtain

$$\begin{aligned} & c(\hat{f}^{\lambda_1}) < c(f^{\lambda_2}) \\ \Leftrightarrow \quad & \sum_{a \in A(G)} c(a) \hat{f}^{\lambda_1}(a) < \sum_{a \in A(G)} c(a) f^{\lambda_2}(a) \\ \Leftrightarrow \quad & \sum_{a \in A^{\text{free}}(G)} c(a) \hat{f}^{\lambda_1}(a) < \sum_{a \in A^{\text{free}}(G)} c(a) f^{\lambda_2}(a) \\ \Leftrightarrow \quad & c(\overline{f}^{\lambda_1}) < c(\tilde{f}^{\lambda_2}). \end{aligned}$$

Furthermore, as by definition flow $\hat{\boldsymbol{f}}$ satisfies the consistent flow constraints, $c(\overline{f}^{\lambda_1}) = \max_{\lambda \in \Lambda} c(\overline{f}^{\lambda})$ is implied by $c(\hat{f}^{\lambda_1}) = \max_{\lambda \in \Lambda} c(\hat{f}^{\lambda})$. Overall, we obtain

$$c(\tilde{f}^{\lambda_2}) > c(\overline{f}^{\lambda_1}) = \max_{\lambda \in \Lambda} c(\overline{f}^{\lambda}) \geq c(\overline{f}^{\lambda_2}),$$

which is a contradiction to the fact that flow $\tilde{f}^{\lambda_2}$ is an optimal $\tilde{b}^{\lambda_2}$-flow for instance $\widetilde{\mathcal{I}}^{\lambda_2}$.

Considering the runtime, the transformation of instance $\mathcal{I}$ to instances $\widetilde{\mathcal{I}}$, $\lambda \in \Lambda$ is done in $\mathcal{O}(|\Lambda| \cdot |A|)$ time. Subsequently, a minimum cost $\tilde{b}^{\lambda}$-flow $\tilde{f}^{\lambda}$ can be computed for every scenario $\lambda \in \Lambda$ by, for example, the Minimum Mean Cycle-Canceling Algorithm in $\mathcal{O}(|A|^3 |V|^2 \log |V|)$ time (Korte et al. 2012). Hence, an optimal robust $\boldsymbol{b}$-flow can be computed in $\mathcal{O}(|\Lambda| \cdot |A|^3 \cdot |V|^2 \log |V|)$ total time. Note that if for a scenario $\lambda \in \Lambda$ no feasible $\tilde{b}^{\lambda}$-flow exists, there also does not exist a robust $\boldsymbol{b}$-flow.

Corollary 2 *The RobMCF≡ problem is solvable in polynomial time for a constant number of fixed arcs.*

Proof We formulate the RobMCF≡ problem as LP where we only require the constant number of variables that indicate the load on the fixed arcs to be integral. The resulting mixed integer linear program (MIP) can be solved in polynomial time by Lenstra's algorithm (1983). In the case that the resulting robust flow is not integral, we define a load by the value of the MIP's integer variables. Analogous to the proof of Lemma 1, we transform the instance and compute an optimal integral robust flow in polynomial time.

At the end of this section, we focus on the objective function of the RobMCF≡ problem. From the MCF problem, or the multi-commodity flow problem (Korte et al. 2012), we know that due to different sources and sinks a flow that sends one unit may cause higher cost than a flow sending two units. Obviously, the same property remains true for instances of the RobMCF≡ problem. If we consider the RobMCF≡ problem on networks with a unique source and a unique sink, we might assume, analogous to the MCF problem, that the cost of a robust flow is determined by the scenario flow which sends the maximum demand. However, the following example shows that this is not true.

Example 2 For a set of two scenarios $\Lambda = \{1, 2\}$, let a network $(G, u, c, \boldsymbol{b})$ with capacity $u \equiv 1$ be given, where digraph G, its cost c, and the non-zero balances $\boldsymbol{b}$ are visualized in Fig. 1b. The only feasible and therefore also optimal solution $\boldsymbol{f} = (f^1, f^2)$ to the RobMCF≡ problem is easy to determine. Considering the second scenario flow f^2 first, the only option to send two flow units from source s to sink t is along paths sv_1t and sv_2t due to the capacity constraints. As the second scenario flow f^2 uses both fixed arcs, the first scenario flow f^1 must also send flow along these arcs. For this reason, the only option to send one flow unit from source s to sink t is along path sv_2v_1t. The cost of the robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ is $c(\boldsymbol{f}) = c(f^1) = 100$.

Corollary 3 *In a network with a unique source and a unique sink, the cost of a robust $\boldsymbol{b}$-flow is not necessarily determined by the b^λ-flow which sends the maximum demand among all scenarios $\lambda \in \Lambda$.*

As a result, independent of the number of sources and sinks given, for solving the RobMCF≡ problem, we cannot only concentrate on a single scenario. However, by reason of the following lemma, in a network with a unique source and a unique sink it is sufficient to concentrate on two scenarios only, namely those in which the minimum and maximum demand is sent.

Lemma 2 *Let $\mathcal{I} = (G = (V, A = A^{\text{free}} \cup A^{\text{fix}}), u, c, \boldsymbol{b})$ be a RobMCF≡ instance with a unique source s and a unique sink t. Without loss of generality, let the scenarios $\lambda \in \Lambda$ be strictly ordered in ascending order of their supply balances b^λ, i.e., $b^1(s) < b^2(s) < \ldots < b^{|\Lambda|}(s)$. Further, let feasible integral b^λ-flows f^λ for scenarios $\lambda = 1$ and $\lambda = |\Lambda|$ be given that satisfy the consistent flow constraints, i.e., $f^1(a) = f^{|\Lambda|}(a)$ for $a \in A^{fix}$. A robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, \ldots, f^{|\Lambda|})$ with cost of $c(\boldsymbol{f}) = \max\{c(f^1), c(f^{|\Lambda|})\}$ can be computed in polynomial time.*

Proof As we consider a network with a unique source and a unique sink, a feasible robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ for instance $\mathcal{I}$ is given by the convex combination of the flows f^1 and $f^{|\Lambda|}$ as follows. For every scenario $\lambda \in \Lambda \setminus \{1, |\Lambda|\}$ let $\gamma^\lambda \in [0, 1]$ be a parameter such that

$$b^\lambda(s) = \gamma^\lambda \cdot b^1(s) + (1 - \gamma^\lambda) \cdot b^{|\Lambda|}(s)$$

holds. We define the corresponding scenario flows $f^\lambda, \lambda \in \Lambda \setminus \{1, |\Lambda|\}$ by

$$f^\lambda(a) := \gamma^\lambda \cdot f^1(a) + (1 - \gamma^\lambda) \cdot f^{|\Lambda|}(a)$$

for all arcs $a \in A$. Flows f^λ, $\lambda \in \Lambda \setminus \{1, |\Lambda|\}$ satisfy the capacity and flow balance constraints, but may be non-integral on some free arcs. Therefore, we restrict every b^λ-flow f^λ, $\lambda \in \Lambda \setminus \{1, |\Lambda|\}$ to the respective MCF instance $\widetilde{\mathcal{I}}^\lambda$ obtained analogous to the Proof of Lemma 1, and this results in feasible $\tilde{b}^\lambda$-flows denoted by $\tilde{f}^\lambda$. Let $\tilde{f}^\lambda_{\text{OPT}}$ be an optimal integral $\tilde{b}^\lambda$-flow for instance $\widetilde{\mathcal{I}}^\lambda$, $\lambda \in \Lambda \setminus \{1, |\Lambda|\}$, then $c(\tilde{f}^\lambda_{\text{OPT}}) \leq c(\tilde{f}^\lambda)$ holds true. For all scenarios $\lambda \in \Lambda \setminus \{1, |\Lambda|\}$, flows $\tilde{f}^\lambda_{\text{OPT}}$ and $\tilde{f}^\lambda$ can be retransformed to flows f^λ_{OPT}, and respectively, f^λ of instance $\mathcal{I}$, ending up in cost of

$$
\begin{aligned}
c(f^{\lambda}_{\mathrm{OPT}}) &= c(\tilde{f}^{\lambda}_{\mathrm{OPT}}) + \sum_{a \in A^{\mathrm{fix}}} c(a) f^{1}(a) \\
&\leq c(\tilde{f}^{\lambda}) + \sum_{a \in A^{\mathrm{fix}}} c(a) f^{1}(a) \\
&= c(f^{\lambda}) \\
&= \gamma^{\lambda} \cdot c(f^{1}) + (1 - \gamma^{\lambda}) \cdot c(f^{|\Lambda|}) \\
&\leq \max\{c(f^{1}), c(f^{|\Lambda|})\}.
\end{aligned}
$$

Consequently, an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f}_{\mathbf{OPT}} = (f^{1}, f^{2}_{\mathrm{OPT}}, \ldots, f^{|\Lambda|-1}_{\mathrm{OPT}}, f^{|\Lambda|})$ with cost $c(\boldsymbol{f}_{\mathbf{OPT}}) = \max\{c(f^{1}), c(f^{|\Lambda|})\}$ is obtained in polynomial time analogous to the Proof of Lemma 1. □

As a result of Lemma 2, if a network with a unique source and a unique sink is given, we only need to concentrate on the first and last scenario to solve the RobMCF≡ problem. We obtain the following conclusion about the problem's complexity which is detailed in the next section.

Corollary 4 *For a set of scenarios Λ with $|\Lambda| \geq 2$, let a RobMCF≡ instance with a unique source and unique sink be given. The complexity is not influenced by the number of scenarios.*

4 Complexity for acyclic digraphs

In this section, we investigate the complexity of the RobMCF≡ problem for networks based on acyclic digraphs. For convenience, we discuss the problem's complexity for networks with a unique source and multiple sinks first. The construction is extended to show the strong $\mathcal{NP}$-completeness for networks with a unique source and a unique sink. Both reductions are performed from the (3, B2)-Sat problem (Berman et al. 2004)—a strongly $\mathcal{NP}$-complete special case of the 3-Sat problem. We start with a formal definition for the decision version of the (3, B2)-Sat problem.

Definition 3 ((3, B2)-Sat) Let $\{x_1, \ldots, x_n\}$ be a set of variables. Further, let $C_1, \ldots, C_m$ be a collection of clauses of size three where every positive and negative literal x_i and $\overline{x}_i$ occur exactly twice. The decision problem of (3, B2)-Sat asks if there exists a variable assignment that satisfies the collection of clauses.

Using the (3, B2)-Sat problem, we obtain the following complexity result.

Theorem 1 *Deciding whether a feasible solution to the RobMCF≡ problem exists for networks based on acyclic digraphs with a unique source but multiple sinks is strongly $\mathcal{NP}$-complete, even if only two scenarios and unit arc capacities are considered.*

For the sake of clarity, we use the notation $[n] := \{1, \ldots, n\}$ in the following.

Proof The RobMCF≡ problem is contained in $\mathcal{NP}$ as we can check in polynomial time whether the flow balance, capacity, and consistent flow constraints are satisfied for every scenario. Further, we show that deciding whether a feasible solution of the RobMCF≡ problem exists is strongly $\mathcal{NP}$-complete by a reduction from the (3, B2)-Sat problem.

Let $\mathcal{I}$ be a (3, B2)-Sat instance with the set of variables $\{x_1, \ldots, x_n\}$ and clauses $C_1, \ldots, C_m$ for which we construct a corresponding RobMCF≡ instance $\widetilde{\mathcal{I}} = (G, u, c, \boldsymbol{b})$

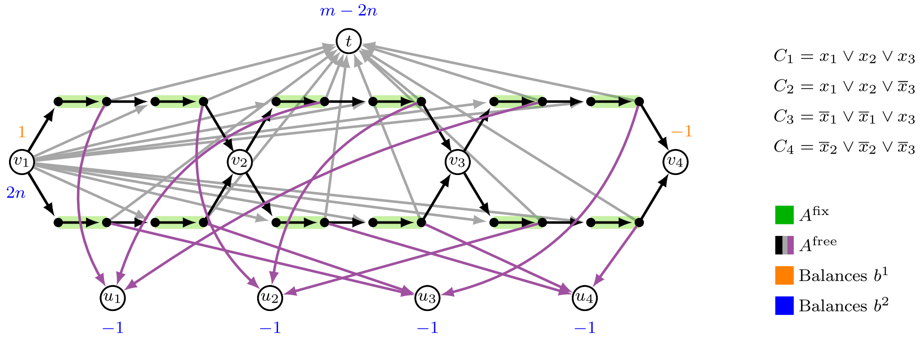
$m-2n$
t
1
v_1
$2n$
v_2
v_3
v_4
-1
u_1
u_2
u_3
u_4
-1
-1
-1
-1
$C_1 = x_1 \vee x_2 \vee x_3$
$C_2 = x_1 \vee x_2 \vee \overline{x}_3$
$C_3 = \overline{x}_1 \vee \overline{x}_1 \vee x_3$
$C_4 = \overline{x}_2 \vee \overline{x}_2 \vee \overline{x}_3$
🟩 A^{fix}
▨ A^{free}
🟧 Balances b^1
🟦 Balances b^2

Fig. 2 Construction of the RobMCF≡ instance $\widetilde{\mathcal{I}}$

considering a set of two scenarios, i.e., $\Lambda = \{1, 2\}$. An example of a RobMCF≡ instance corresponding to a (3, B2)-Sat instance with four clauses and three variables is visualized in Fig. 2. In general, the instance is based on a digraph $G = (V, A)$ defined as follows. The vertex set V consists of one vertex v_i per variable x_i, $i \in [n]$, an additional dummy vertex v_{n+1}, and one vertex u_j per clause C_j, $j \in [m]$. In addition, for every literal x_i ($\overline{x}_i$), $i \in [n]$ four auxiliary vertices w_i^ℓ ($\overline{w}_i^\ell$), $\ell \in [4]$ are included as well as a further auxiliary vertex t. Arc set A includes arcs that connect two successive variable vertices v_i, v_{i+1}, $i \in [n]$ by two parallel paths p_i and $\overline{p}_i$ defined along the auxiliary vertices, i.e., $p_i = v_i w_i^1 w_i^2 w_i^3 w_i^4 v_{i+1}$ and $\overline{p}_i = v_i \overline{w}_i^1 \overline{w}_i^2 \overline{w}_i^3 \overline{w}_i^4 v_{i+1}$ for $i \in [n]$. Path p_i represents the positive literal x_i, and path $\overline{p}_i$ the negative literal $\overline{x}_i$ of instance $\mathcal{I}$. As each literal occurs exactly twice in instance $\mathcal{I}$, we identify two arcs of paths p_i and $\overline{p}_i$ each with the literals. More precisely, let x_i^k ($\overline{x}_i^k$) denote literal x_i ($\overline{x}_i$) which occurs the k-th time, $k \in [2]$ in the formula. Arc (w_i^{2k-1}, w_i^{2k}) $((\overline{w}_i^{2k-1}, \overline{w}_i^{2k}))$, which we refer to as *literal arc*, is supposed to correspond to literal x_i^k ($\overline{x}_i^k$). Using this correspondence, we add arc (w_i^{2k}, u_j) $((\overline{w}_i^{2k}, u_j))$ for every literal x_i^k ($\overline{x}_i^k$), $i \in [n]$, $k \in [2]$ included in clause C_j, $j \in [m]$. Finally, arcs (v_1, w_i^ℓ) and $(v_1, \overline{w}_i^\ell)$ for $\ell \in \{1, 3\}$ as well as arcs (w_i^ℓ, t) and $(\overline{w}_i^\ell, t)$ for $\ell \in \{2, 4\}$ are added for every $i \in [n]$.

The fixed arcs of set A are defined by all literal arcs, i.e., $A^{\text{fix}} = \big\{(w_i^\ell, w_i^{\ell+1}), (\overline{w}_i^\ell, \overline{w}_i^{\ell+1}) \mid \ell \in \{1, 3\},\ i \in [n]\big\}$, while all remaining arcs are contained in set A^{free}. We set the capacity and cost to $u \equiv 1$ and $c \equiv 0$, respectively. To conclude, we define balances $\boldsymbol{b} = (b^1, b^2)$ such that the unique source is given by vertex v_1. In contrast, depending on the scenario considered, vertex v_{n+1}, or vertices $u_1, \ldots, u_m$ and t function as sinks. More precisely, we obtain

$$b^1(v) = \begin{cases} 1 & \text{if } v = v_1, \\ -1 & \text{if } v = v_{n+1}, \\ 0 & \text{otherwise}, \end{cases} \quad b^2(v) = \begin{cases} 2n & \text{if } v = v_1, \\ -1 & \text{if } v = u_j,\ \ j \in [m], \\ m-2n & \text{if } v = t, \\ 0 & \text{otherwise}. \end{cases}$$

In summary, we obtain a feasible RobMCF≡ instance $\widetilde{\mathcal{I}} = (G, u, c, \boldsymbol{b})$ that can be constructed in polynomial time. Hence, it remains to show that $\mathcal{I}$ is a Yes-instance if and only if for instance $\widetilde{\mathcal{I}}$ a feasible robust $\boldsymbol{b}$-flow exists.

For this purpose, let $x_1, \dots, x_n$ be a satisfying truth assignment for instance $\mathcal{I}$. We define the first scenario flow f^1 of instance $\widetilde{\mathcal{I}}$ as follows

$$f^1(a) = \begin{cases} 1 \text{ for all } a \in A(p_i) \text{ if } x_i = \textsc{True}, \\ 1 \text{ for all } a \in A(\overline{p}_i) \text{ if } x_i = \textsc{False}, \\ 0 \text{ otherwise,} \end{cases}$$

i.e., flow f^1 sends exactly one unit from source v_1 to sink v_{n+1} using $2n$ literal arcs by sending flow along either path p_i or $\overline{p}_i$, $i \in [n]$. As $x_1, \dots, x_n$ is a satisfying truth assignment, there exists one verifying literal x_i^k or $\overline{x}_i^k$, $k \in [2]$, $i \in [n]$ for each clause C_j, $j \in [m]$. We define the second scenario flow f^2 from the source to the clause vertices along the literal arcs which correspond to these verifying literals:

$$f^2(a) = \begin{cases} 1 & \text{for all } a \in A(q) \text{ with } q = v_1 w_i^{2k-1} w_i^{2k} u_j \text{ if } x_i^k \in C_j \text{ is verifying,} \\ 1 & \text{for all } a \in A(q) \text{ with } q = v_1 \overline{w}_i^{2k-1} \overline{w}_i^{2k} u_j \text{ if } \overline{x}_i^k \in C_j \text{ is verifying,} \\ 0 & \text{otherwise.} \end{cases}$$

To satisfy the remaining $2n - m$ demand, flow f^2 is defined along the remaining, and also from flow f^1 used, literal arcs to sink t, i.e., $f^2(a) = 1$ for all

$$\begin{cases} a \in A(p) \text{ with } p = v_1 w_i^{2k-1} w_i^{2k} t \text{ if } x_i = \textsc{True} \text{ and } f^2((w_i^{2k-1}, w^{2k})) = 0, \\ a \in A(p) \text{ with } p = v_1 \overline{w}_i^{2k-1} \overline{w}_i^{2k} t \text{ if } x_i = \textsc{False} \text{ and } f^2((\overline{w}_i^{2k-1}, \overline{w}^{2k})) = 0. \end{cases}$$

Overall, flow f^2 sends $2n - m$ units to sink t and one unit to each of the sinks $u_1, \dots, u_m$ using exactly $2n$ literal arcs. Consequently, we have constructed b^λ-flows f^λ for both scenarios $\lambda \in \Lambda$ such that the consistent flow constraints are satisfied, and this results in a feasible robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$.

Conversely, let $\boldsymbol{f} = (f^1, f^2)$ be a feasible robust $\boldsymbol{b}$-flow. Flow f^2 sends in total $2n$ units from vertex v_1 to vertices $u_1, \dots, u_m$ and t. By construction of the network, the only option to reach each of these sinks requires the usage of at least one of the fixed literal arcs. Due to the integral flow f^1 sending only one unit within the acyclic digraph, it holds $f^1(a) = f^2(a) \in \{0, 1\}$ for all fixed arcs $a \in A^{\text{fix}}$. Consequently, flows f^1 and f^2 use at least $2n$ fixed arcs in order to meet the demand of flow f^2. As further consequence of the acyclic digraph, flow f^1 uses either path p_i or $\overline{p}_i$, $i \in [n]$ but never both simultaneously to reach sink v_{n+1}. Accordingly, if flow f^1 sends flow along path p_i, $i \in [n]$, we set $x_i = \textsc{True}$. If flow f^1 sends flow along path $\overline{p}_i$ for $i \in [n]$, our choice is $x_i = \textsc{False}$. Further, to meet the demand at sinks u_j, $j \in [m]$, flow f^2 sends flow along either subpath $w_i^\ell w_i^{\ell+1} u_j$ or $\overline{w}_i^\ell \overline{w}_i^{\ell+1} u_j$ for $\ell \in \{1, 3\}$, $i \in [n]$, $j \in [m]$ but never both simultaneously. In the former case, clause C_j is verified due to the previous assignment $x_i = \textsc{True}$ induced by flow f^1 and the fact that $x_i \in C_j$ holds. In the latter case, clause C_j is verified due to the included variable x_i set to $\textsc{False}$. As a result, $x_1, \dots, x_n$ is a satisfying truth assignment for instance $\mathcal{I}$.

The statement of Theorem 1 can be formulated even stronger as the following theorem shows.

Theorem 2 *Deciding whether a feasible solution to the RobMCF$\equiv$ problem exists for networks based on acyclic digraphs is strongly $\mathcal{NP}$-complete, even if only two scenarios on a network with a unique source, a unique sink, and unit arc capacities are considered.*

Proof We extend the construction of Proof of Theorem 1 by free arcs (u_j, v_{n+1}) for $j \in [m]$ as well as the two free arcs (v_1, w_n^4) and $(v_1, \overline{w}_n^4)$. Like all other arcs in the network, their capacities and costs are set to one and zero, respectively. The balances are updated such that v_1 serves as unique source and v_{n+1} as unique sink. However, in the second scenario we require $2n + 2$ instead of $2n$ demand sent from the source to the sink. This adjustment is necessary as we need to ensure that sufficient demand is sent along the clause vertices which are no sinks anymore. Otherwise, there might exist a feasible robust $\boldsymbol{b}$-flow that sends a unit along path $v_1 w_n^3 w_n^4 v_{n+1}$ or $v_1 \overline{w}_n^3 \overline{w}_n^4 v_{n+1}$ which in turn allows one unsatisfied clause. Analogous to the Proof of Theorem 1, a $(3, B2)$-SAT instance is a Yes-instance if and only if for the corresponding RobMCF≡ instance a feasible robust flow exists.

5 RobMCF≡ problem on series-parallel digraphs

In this section, we consider the RobMCF≡ problem on series-parallel (SP) digraphs. We firstly present a definition of SP digraphs and their representation in the form of a rooted binary decomposition tree. Exploiting the decomposition tree, i.e., the instruction of the composition of an SP digraph, we reduce the complexity for two special cases of the RobMCF≡ problem, unless $\mathcal{P} = \mathcal{NP}$. In Sect. 5.1, we show the weak $\mathcal{NP}$-completeness for networks with multiple sources and multiple sinks. For the special case of a constant number of scenarios, we exclude the strong $\mathcal{NP}$-completeness by proposing a pseudo-polynomial time algorithm. For the special case of networks with a unique source and a unique sink, we provide an algorithm which runs in polynomial time as shown in Sect. 5.2.

We start with a formal definition for SP digraphs based on the edge SP multi-graphs definition of Valdes et al. (1982).

Definition 4 *Series-parallel (SP) digraphs* can be recursively defined as follows.

1. An arc (o, q) is an SP digraph with *origin* o and *target* q.
2. Let G_1 with origin o_1 and target q_1 and G_2 with origin o_2 and target q_2 be SP digraphs. The digraph that is constructed by one of the following two compositions of SP digraphs G_1 and G_2 is itself an SP digraph.
   (a) The *series composition* G of two SP digraphs G_1 and G_2 is the digraph obtained by contracting target q_1 and origin o_2. The origin of digraph G is then o_1 (becoming o), and the target is q_2 (becoming q).
   (b) The *parallel composition* G of two SP digraphs G_1 and G_2 is the digraph obtained by contracting origins o_1 and o_2 (becoming o) and contracting targets q_1 and q_2 (becoming q). The new origin of digraph G is o, and the target is q.

The parallel and series compositions are illustrated in Fig. 3a. Note that, by definition, SP digraphs are generally multi-graphs with one definite origin and one definite target.

A useful property of SP digraphs is their representability in the form of a rooted binary decomposition tree, a so-called *SP tree*, visualized in Fig. 3b. For a given SP digraph, we construct an SP tree that represents the order of the series and parallel compositions of individual arcs. By means of three different vertices, namely L-vertices, S-vertices, and P-vertices, single arcs as well as series and parallel compositions are indicated. The SP tree's leaves are L-vertices where there exist as many L-vertices as the digraph represented has arcs. The S- and P-vertices are the SP tree's inner vertices and correspond to the digraph obtained by a series or, respectively, parallel composition of the subgraphs associated with their two child vertices. The order of the children of P-vertices is irrelevant while it is essential for S-vertices as the series composition is not commutative. Following the constructions of all

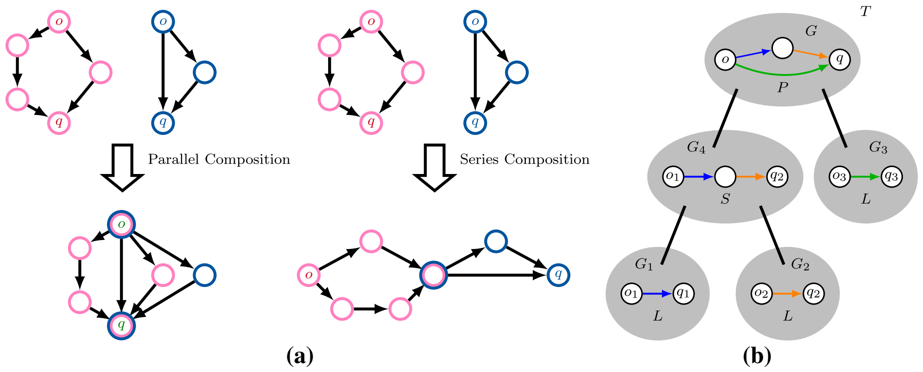

Fig. 3 **a** Example of an SP digraph defined by a parallel or series composition **b** Example of the representation of an SP digraph by its SP tree

series and parallel compositions, we obtain the entire digraph represented by the SP tree's root. The representation of an SP digraph by its SP tree can be beneficially used as the construction is conducted in polynomial time (Valdes et al. 1979).

Using the SP trees, we define pearl digraphs—SP digraphs with a special structure—based on the definition of Ohst (2016).

Definition 5 *Pearl digraphs* are SP digraphs whose SP trees do not have P-vertices whose children in turn are S-vertices.

In other words, a pearl digraph is a digraph that consists of a path with multi-arcs.

5.1 Multiple sources and multiple sinks networks

In this section, we firstly concentrate on the complexity of the RobMCF≡ problem for networks based on SP digraphs. For the special case that the number of scenarios is constant, we propose a pseudo-polynomial time algorithm. We start with a formal definition for the decision version of the RobMCF≡ problem.

Definition 6 The decision version of the RobMCF≡ problem asks whether a robust flow exists with cost at most $\beta \in \mathbb{Z}_{\geq 0}$.

We perform a reduction from Partition, which is known to be weakly $\mathcal{NP}$-complete (Johnson and Garey 1979).

Definition 7 (Partition) Let $S = \{s_1, \ldots, s_n\}$ be a set of n positive integers that sum up to $2w$, i.e., $\sum_{i=1}^{n} s_i = 2w$. The decision problem of Partition asks whether there exists a partition of set S in two disjoint subsets S_1 and S_2 such that the sum of the integers of subsets S_1 is equal to the sum of the integers included in subset S_2, i.e., $S = S_1 \uplus S_2$ with

$$\sum_{s_i \in S_1} s_i = \sum_{s_i \in S_2} s_i = w.$$

Theorem 3 *The decision version of the RobMCF≡ problem on networks based on SP digraphs is weakly $\mathcal{NP}$-complete, even if only two scenarios, unit arc capacities, a unique source, and single sinks are considered on pearl digraphs.*

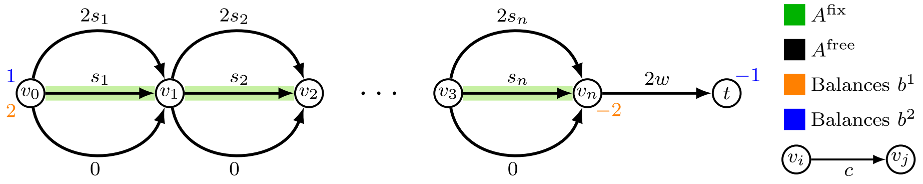

Fig. 4 Construction of the RobMCF≡ instance $\widetilde{\mathcal{I}}$

Proof Let $\mathcal{I}$ be a Partition instance with positive integers $s_1, \ldots, s_n$ such that $\sum_{i=1}^{n} s_i = 2w$ holds. We construct a corresponding RobMCF≡ instance $\widetilde{\mathcal{I}} = (G, u, c, \boldsymbol{b})$ considering a set of two scenarios, i.e., $\Lambda = \{1, 2\}$, visualized in Fig. 4. The network is based on a pearl digraph $G = (V, A)$ where vertex set V consists of two auxiliary vertices v_0 and t, and one vertex v_i per integer s_i, $i \in [n]$. Arc set A consists of multi-arcs a_i for $i \in [n]$ that connects two successive vertices v_{i-1}, v_i by three parallel arcs a_i^1, a_i^2, a_i^3, plus a single arc a_{n+1} from vertex v_n to vertex t. Multi-arc a_i, $i \in [n]$ is supposed to represent integer s_i, which is why we refer to as *integer multi-arcs*. The fixed arcs of set A are defined by the second arc of all integer multi-arcs each, i.e., $A^{\text{fix}} = \{a_i^2 = (v_{i-1}, v_i) \mid i \in [n]\}$, while all remaining arcs are contained in set A^{free}.

Further, we set the capacity on all arcs to one, i.e., $u \equiv 1$. The cost c is given such that the use of the first arc a_i^1 and second arc a_i^2 of every integer multi-arc costs two and one times the corresponding integer value s_i per flow unit, respectively. In turn, using the third arc a_i^3 causes zero cost. The use of arc a_{n+1} costs $2w$ per flow unit. To conclude, we define balances $\boldsymbol{b} = (b^1, b^2)$ on vertex set V such that in the first scenario vertex v_0 supplies and vertex v_n demands two units. In the second scenario, vertex v_0 supplies and vertex t demands one unit. In both scenarios, the balances of all other vertices are equal to zero, i.e., overall we obtain

$$b^1(v) = \begin{cases} 2 & \text{if } v = v_0, \\ -2 & \text{if } v = v_n, \\ 0 & \text{otherwise,} \end{cases} \quad b^2(v) = \begin{cases} 1 & \text{if } v = v_0, \\ -1 & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases}$$

Accordingly, for both scenarios the unique source is defined by vertex v_0 while depending on the scenario considered vertex v_n or t serves as single sink. In order to satisfy demand with supply, flow is sent along paths through the network. For convenience, let p^ℓ denote the path along the ℓ-th integer multi-arcs for $\ell \in [3]$, i.e., $p^\ell = a_1^\ell \ldots a_n^\ell$. Overall, we obtain a feasible RobMCF≡ instance $\widetilde{\mathcal{I}} = (G, u, c, \boldsymbol{b})$ that can be constructed in polynomial time. Hence, it remains to show that $\mathcal{I}$ is a Yes-instance if and only if for instance $\widetilde{\mathcal{I}}$ a robust $\boldsymbol{b}$-flow exists with cost of at most $\beta := 3w$.

For this purpose, let S_1 and S_2 be a feasible partition for instance $\mathcal{I}$. We define the first scenario flow f^1 of instance $\widetilde{\mathcal{I}}$ by

$$f^1(a) = \begin{cases} 1 \text{ for all arcs } a = a_i^1 \in A \text{ if } s_i \in S_1, \\ 1 \text{ for all arcs } a = a_i^2 \in A \text{ if } s_i \in S_2, \\ 1 \text{ for all arcs } a = a_i^3 \in A, \\ 0 \text{ otherwise,} \end{cases}$$

i.e., flow f^1 sends one unit from source v_0 to sink v_n along arcs of paths p^1 and p^2, while one further unit is sent using arcs of path p^3 only. As the sets S_1 and S_2 form a feasible partition we obtain cost of

$$c(f^1) = \sum_{a \in A} c(a) f^1(a) = \sum_{a \in A^{\text{fix}}} c(a) f^1(a) + \sum_{a \in A^{\text{free}}} c(a) f^1(a) = w + 2w = 3w.$$

According to flow f^1 and the partition, we define the second scenario flow f^2 by

$$f^2(a) = \begin{cases} 1 \text{ for all arcs } a = a_i^2 \in A \text{ if } s_i \in S_2, \\ 1 \text{ for all arcs } a = a_i^3 \in A \text{ if } s_i \in S_1, \\ 1 \text{ for arc } a = a_{n+1} \in A, \\ 0 \text{ otherwise,} \end{cases}$$

i.e., flow f^2 sends exactly one unit from source v_0 to sink t along arcs of paths p^2 and p^3, and by using arc a_{n+1}. The following cost is caused

$$\begin{aligned} c(f^2) &= \sum_{a \in A} c(a) f^2(a) \\ &= \sum_{a \in A^{\text{fix}}} c(a) f^2(a) + \sum_{a \in A^{\text{free}} \setminus \{a_{n+1}\}} c(a) f^2(a) + c(a_{n+1}) f^2(a_{n+1}) \\ &= w + 0 + 2w = 3w. \end{aligned}$$

Consequently, we have constructed a robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ with cost of $3w$.

Conversely, let $\boldsymbol{f} = (f^1, f^2)$ be a robust $\boldsymbol{b}$-flow with cost of at most $3w$, i.e., $c(\boldsymbol{f}) = \max\{c(f^1), c(f^2)\} \leq 3w$. The first scenario flow f^1 sends two units from source v_0 to sink v_n. Due to the capacities, not only a single path is used to send these flow units. In particular, not only path p^3 causing zero cost is used. As sending one flow unit along path p^1 would cause cost of

$$c(p^1) = \sum_{a \in A(p^1)} c(a) = \sum_{i=1}^{n} 2s_i = 4w > 3w \geq \max\{c(f^1), c(f^2)\},$$

flow f^1 does not use all arcs of path p^1 either. Accordingly, flow f^1 uses as many arcs of path p^2 as at least cost of w is caused in order that at most $2w$ cost is caused due to arcs of path p^1.

The second scenario flow f^2 sends one unit from source v_0 to sink t. As flow f^2 uses arc a_n with cost of $2w$ to reach sink t, the unit is sent along arcs of paths p^1, p^2, p^3 such that at most cost of w is caused. Furthermore, as flow f^1 uses as many arcs of path p^2 as at least cost of w is caused, this also holds true for flow f^2 as $A(p^2) = A^{\text{fix}}$ holds. Consequently, flow f^2 only uses arcs of path p^2 and p^3, however, due to the acyclic pearl digraph never of the same multi-arc simultaneously such that the sets

$$\begin{aligned} S_1 &:= \{s_i \mid f^2(a_i^2) = 1 \text{ for arc } a_i^2 \in A^{\text{fix}} \text{ with } i \in [n]\}, \\ S_2 &:= \{s_i \mid f^2(a_i^3) = 1 \text{ for arc } a_i^3 \in A(p^3) \text{ with } i \in [n]\} \end{aligned}$$

form a feasible partition for instance $\mathcal{I}$.

In the next step, we refute the strong $\mathcal{NP}$-completeness in the special case of a constant number of scenarios. Therefore, we propose a pseudo-polynomial algorithm based on dynamic programming. The dynamic program (DP) is applicable for networks with an arbitrary number of sources and sinks, especially for multiple sources and multiple sinks. The core idea of the DP is a bottom-up method using the SP tree. While composing the SP digraph step by step, in each of these steps a robust flow is sought satisfying additional restrictions explained in the following. The flow needs to send a given supply from the origin through the subgraph considered in the current step. Further, the flow needs to satisfy the inner vertices' balances as their in- and outgoing arcs are already set. In contrast, the balances at the origin and target do not have to be satisfied as in subsequent steps further subgraphs can still be composed at these vertices. Moreover, the flow must exactly meet a budget given. Backtracking the steps of the DP results to an optimal robust flow. Before we present the DP in more detail, we introduce the notations and labels needed.

Let us consider a RobMCF≡ instance $(G, u, c, \boldsymbol{b})$ where G is an SP digraph with origin o and target q. Further, let T be the SP tree of digraph G with its root vertex $r \in V(T)$. We denote the subgraph of digraph G that is associated to vertex $v \in V(T)$ by G_v, and its origin and target by o_v and q_v, respectively. The algorithm relies on demand labels $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ defined for every subgraph G_v associated with a vertex $v \in V(T)$. The parameter vector $\tilde{\boldsymbol{s}}_{\boldsymbol{v}} = (\tilde{s}_v^1, \ldots, \tilde{s}_v^{|\Lambda|}) \in \mathbb{Z}_{\geq 0}^{|\Lambda|}$ determines for all scenarios the supply at origin o_v of subgraph G_v. For every scenario $\lambda \in \Lambda$ the supply $\tilde{s}_v^\lambda$ is limited by the sum of the capacities of all outgoing arcs of origin o_v of subgraph G_v, i.e., $\tilde{s}_v^\lambda \in \{0, \ldots, U_v\}$ with $U_v = \sum_{a=(o_v,w)\in A(G_v)} u(a)$. The parameter vector $\tilde{\boldsymbol{c}}_{\boldsymbol{v}} = (\tilde{c}_v^1, \ldots, \tilde{c}_v^{|\Lambda|}) \in \mathbb{Z}_{\geq 0}^{|\Lambda|}$ specifies for all scenarios the budget that must be spent for sending the supply in subgraph G_v with respect to cost function c. Consequently, an upper bound on the budget is given by the cost that may occur in subgraph G_v, i.e., $\tilde{c}_v^\lambda \in \{0, \ldots, C_v\}$ for $\lambda \in \Lambda$ with $C_v = \sum_{a\in A(G_v)} c(a) \cdot u(a)$.

Let the $(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$-restricted robust minimum cost flow problem under consistent flow constraints (rRobMCF≡ $(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$) be defined as the RobMCF≡ problem on subgraph G_v, $v \in V(T)$ with restrictions implied by supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}}$ and budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}}$. The demand label $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ is defined as the optimal solution value of the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem. For convenience and for the sake of clarity, we indicate the rRobMCF≡ $(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem by the following integer program formulation.

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = \min \ 0 \tag{1}$$

$$\text{s.t.} \sum_{a\in A(G_v)} c(a) \cdot f_a^\lambda = \tilde{c}_v^\lambda \qquad \forall \lambda \in \Lambda \tag{2}$$

$$\sum_{a=(w,z)\in A(G_v)} f_a^\lambda - \sum_{a=(z,w)\in A(G_v)} f_a^\lambda = \begin{cases} b^\lambda(w) & \text{if } w \neq o_v, \\ \tilde{s}_w^\lambda & \text{if } w = o_v, \end{cases} \qquad \forall w \in V(G_v) \setminus \{q_v\}, \lambda \in \Lambda \tag{3}$$

$$f_a^\lambda = f_a^{\lambda'} \qquad \forall a \in A^{\text{fix}}(G_v), \lambda, \lambda' \in \Lambda \tag{4}$$

$$0 \leq f_a^\lambda \leq u(a) \qquad \forall a \in A(G_v), \lambda \in \Lambda \tag{5}$$

$$f_a^\lambda \in \mathbb{Z}_{\geq 0} \qquad \forall a \in A(G_v), \lambda \in \Lambda \tag{6}$$

The rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem requires a robust $\boldsymbol{b}$-flow in subgraph G_v by means of constraints (3)–(6). Therefore, the flow needs to satisfy the supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}}$ at origin o_v, and the balances $\boldsymbol{b}$ at all other vertices except target q_v. Furthermore, the flow must exactly meet the budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}}$, see constraint (2). By definition of the objective function (1), finding a feasible solution is sufficient to solve the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem, i.e., $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) \in \{0, \infty\}$.

For solving the RobMCF≡ problem on SP digraphs, the DP exploits the structure of the SP tree to compute demand labels recursively. More precisely, considering a specific vertex in the SP tree, we update the corresponding demand label based on the labels corresponding to the children's vertices in a bottom-up procedure. Depending on whether the SP tree's vertex considered is an L-, S-, or P-vertex, one of the following three procedures is applied. An example of the procedures of the DP is given in Appendix A. We start with the initialization at the leaves.

Lemma 3 *Let $v \in V(T)$ be a leaf of SP tree T, i.e., v is an L-vertex. The demand label $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ is initialized by*

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = \begin{cases} 0 & \textit{if } (o_v, q_v) \in A^{\text{free}}(G_v),\ \tilde{c}_v^\lambda = c((o_v, q_v)) \cdot \tilde{s}_v^\lambda,\ \forall \lambda \in \Lambda, \\ 0 & \textit{if } (o_v, q_v) \in A^{\text{fix}}(G_v),\ \tilde{s}_v^\lambda = \tilde{s}_v^{\lambda'},\ \tilde{c}_v^\lambda = c((o_v, q_v)) \cdot \tilde{s}_v^\lambda,\ \forall \lambda, \lambda' \in \Lambda, \\ \infty & \textit{otherwise}. \end{cases}$$

Proof As $v \in V(T)$ is an L-vertex, subgraph G_v only consists of the single arc $a_v := (o_v, q_v)$. If a_v is a free arc, i.e., $a_v \in A^{\text{free}}(G_v)$, $\tilde{c}_v^\lambda = c((o_v, q_v)) \cdot \tilde{s}_v^\lambda$ must hold true. Otherwise, there exists no feasible flow that satisfies constraints (2) due to constraints (3). Consequently, the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem is not solvable, i.e., $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = \infty$. If a_v is a fixed arc, i.e., $a_v \in A^{\text{fix}}(G_v)$, the constraints of the previous case need to be satisfied due to the same argumentation. In addition, $\tilde{s}_v^\lambda = \tilde{s}_v^{\lambda'}$ must hold true for all scenarios $\lambda, \lambda' \in \Lambda$ by reason of constraints (4). To conclude, if the presented constraints are satisfied, an optimal solution to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem is given by $\boldsymbol{f}(a_v) := \tilde{\boldsymbol{s}}_{\boldsymbol{v}}$ such that $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = 0$ holds true.

In the next step, we consider the case in which the demand label is derived recursively from the demand labels of the child vertices that are parallelly composed.

Lemma 4 *Let $v \in V(T)$ be a P-vertex in SP tree T with child vertices $x, y \in V(T)$. The demand label $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ at vertex v can be computed by a composition of the demand labels $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ of its child vertices x and y as follows*

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = \min_{\substack{\tilde{\boldsymbol{s}}_{\boldsymbol{v}} = \tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{y}} \\ \tilde{\boldsymbol{c}}_{\boldsymbol{v}} = \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\}.$$

Proof For vertex $v \in V(T)$, let $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ be the demand label with the related solution $\boldsymbol{f}^*$. As v is a P-vertex, flow $\boldsymbol{f}^*$ with associated supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}} = \sum_{a=(o_v,w) \in A(G_v)} \boldsymbol{f}^*(a)$ can be divided into two flows $\boldsymbol{f}_{\boldsymbol{x}}$ and $\boldsymbol{f}_{\boldsymbol{y}}$ with associated supplies $\tilde{\boldsymbol{s}}_{\boldsymbol{x}}$ and $\tilde{\boldsymbol{s}}_{\boldsymbol{y}}$, respectively. Flow $\boldsymbol{f}_{\boldsymbol{x}}$ is defined on subgraph G_x, and flow $\boldsymbol{f}_{\boldsymbol{y}}$ is defined on subgraph G_y only. The budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}} = \sum_{a \in A(G_v)} c(a) \cdot \boldsymbol{f}^*(a)$ of flow $\boldsymbol{f}^*$ is also divided such that $\tilde{\boldsymbol{c}}_{\boldsymbol{x}}$ describes the budget of flow $\boldsymbol{f}_{\boldsymbol{x}}$ and $\tilde{\boldsymbol{c}}_{\boldsymbol{y}}$ the budget of flow $\boldsymbol{f}_{\boldsymbol{y}}$. Flows $\boldsymbol{f}_{\boldsymbol{x}}$ and $\boldsymbol{f}_{\boldsymbol{y}}$ are feasible solutions to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ problem, respectively. Consequently, we obtain

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \geq d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}),$$

where $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ are the demand labels corresponding to the child vertices $x, y \in V(T)$. In particular, this implies

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) \geq \min_{\substack{\tilde{\boldsymbol{s}}_{\boldsymbol{v}} = \tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{y}} \\ \tilde{\boldsymbol{c}}_{\boldsymbol{v}} = \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\}.$$

Conversely, for child vertices $x, y \in V(T)$, let $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ be the demand labels with related solutions $\boldsymbol{f}_{\boldsymbol{x}}^*$ and $\boldsymbol{f}_{\boldsymbol{y}}^*$. Combining flows $\boldsymbol{f}_{\boldsymbol{x}}^*$ and $\boldsymbol{f}_{\boldsymbol{y}}^*$ results in a feasible solution $\boldsymbol{f}_{\boldsymbol{v}} := \boldsymbol{f}_{\boldsymbol{x}}^* + \boldsymbol{f}_{\boldsymbol{y}}^*$ to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem with supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}} := \tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{y}}$ and budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}} := \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{y}$. Consequently, for all supplies $\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{s}}_{\boldsymbol{y}}$ and budgets $\tilde{\boldsymbol{c}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}$ given the following holds true

$$d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \geq d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) = d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}),$$

where $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ is the demand label corresponding to vertex $v \in V(T)$. This implies

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) \leq \min_{\substack{\tilde{\boldsymbol{s}}_v = \tilde{\boldsymbol{s}}_x + \tilde{\boldsymbol{s}}_y \\ \tilde{\boldsymbol{c}}_v = \tilde{\boldsymbol{c}}_x + \tilde{\boldsymbol{c}}_y}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\}.$$

To conclude the computation of demand labels, we consider the case in which a demand label is derived recursively from the demand labels of the child vertices that are serially composed.

Lemma 5 *Let $v \in V(T)$ be an S-vertex in SP tree T with child vertices $x, y \in V(T)$. The demand label $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ at vertex v can be computed by a composition of the demand labels $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ of its child vertices x and y by*

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = \min_{\substack{\tilde{\boldsymbol{s}}_x = \tilde{\boldsymbol{s}}_v \\ \tilde{\boldsymbol{s}}_y = \tilde{\boldsymbol{s}}_x + \boldsymbol{\beta} \\ \tilde{\boldsymbol{c}}_v = \tilde{\boldsymbol{c}}_x + \tilde{\boldsymbol{c}}_y}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\},$$

where $\boldsymbol{\beta} = (\beta^1, \ldots, \beta^{|\Lambda|})$ with $\beta^\lambda := \sum_{v \in V(G_x)\setminus\{o_x\}} b^\lambda(v)$ holds for every scenario $\lambda \in \Lambda$.

Proof For vertex $v \in V(T)$, let $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ be the demand label with the related solution $\boldsymbol{f}^*$. We assume that digraph G_v is constructed by contracting the target q_x of subgraph G_x with the origin o_y of subgraph G_y. Consequently, the flow that is sent through subgraph G_y requires on the one hand the access via origin o_y. On the other hand, at least the same amount of flow is originated in subgraph G_x in the first place. Using this insight, we partition flow $\boldsymbol{f}^*$ and the associated supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}} = \sum_{a=(o_v,w)\in A(G_v)} \boldsymbol{f}^*(a)$ in two flows $\boldsymbol{f}_{\boldsymbol{x}}$ and $\boldsymbol{f}_{\boldsymbol{y}}$ where flow $\boldsymbol{f}_{\boldsymbol{x}}$ is defined on subgraph G_x and flow $\boldsymbol{f}_{\boldsymbol{y}}$ is defined on subgraph G_y only. More precisely, we obtain $\boldsymbol{f}_{\boldsymbol{x}}(a) := \boldsymbol{f}^*(a)$ for all arcs $a \in A(G_x)$ with associated supply $\tilde{\boldsymbol{s}}_{\boldsymbol{x}} = \tilde{\boldsymbol{s}}_{\boldsymbol{v}}$, and $\boldsymbol{f}_{\boldsymbol{y}}(a) := \boldsymbol{f}^*(a)$ for all arcs $a \in A(G_y)$ with associated supply $\tilde{\boldsymbol{s}}_{\boldsymbol{y}} := \sum_{a=(o_y,w)\in A(G_y)} \boldsymbol{f}^*(a) = \tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \boldsymbol{\beta}$ where $\boldsymbol{\beta} = (\beta^1, \ldots, \beta^{|\Lambda|})$ with $\beta^\lambda := \sum_{v\in V(G_x)\setminus\{o_x\}} b^\lambda(v)$, $\lambda \in \Lambda$. The associated supply $\tilde{\boldsymbol{s}}_{\boldsymbol{y}}$ results from the supply $\tilde{\boldsymbol{s}}_{\boldsymbol{x}}$ plus the flow that originates from sources (that are different from the origin) in subgraph G_x minus the flow that is absorbed at sinks in subgraph G_x. The budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}} = \sum_{a\in A(G_v)} c(a) \cdot \boldsymbol{f}^*(a)$ of flow $\boldsymbol{f}^*$ can also be divided such that $\tilde{\boldsymbol{c}}_{\boldsymbol{x}}$ describes the budget of flow $\boldsymbol{f}_{\boldsymbol{x}}$ and $\tilde{\boldsymbol{c}}_{\boldsymbol{y}}$ the one of flow $\boldsymbol{f}_{\boldsymbol{y}}$. Flows $\boldsymbol{f}_{\boldsymbol{x}}$ and $\boldsymbol{f}_{\boldsymbol{y}}$ are feasible solutions to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ problem, respectively. Consequently, we obtain

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) = d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \geq d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}),$$

where $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ are the demand labels corresponding to child vertices $x, y \in V(T)$. In particular, this implies

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) \geq \min_{\substack{\tilde{\boldsymbol{s}}_x = \tilde{\boldsymbol{s}}_v \\ \tilde{\boldsymbol{s}}_y = \tilde{\boldsymbol{s}}_x + \boldsymbol{\beta} \\ \tilde{\boldsymbol{c}}_v = \tilde{\boldsymbol{c}}_x + \tilde{\boldsymbol{c}}_y}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\}.$$

Conversely, for child vertices $x, y \in V(T)$, let $d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}})$ and $d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}})$ with $\tilde{\boldsymbol{s}}_{\boldsymbol{y}} = \tilde{\boldsymbol{s}}_{\boldsymbol{x}} + \boldsymbol{\beta}$ be the demand labels with related solutions $\boldsymbol{f}_{\boldsymbol{x}}^*$ and $\boldsymbol{f}_{\boldsymbol{y}}^*$. Combining flows $\boldsymbol{f}_{\boldsymbol{x}}^*$ and $\boldsymbol{f}_{\boldsymbol{y}}^*$ results in a feasible solution $\boldsymbol{f}_{\boldsymbol{v}} := \boldsymbol{f}_{\boldsymbol{x}}^* + \boldsymbol{f}_{\boldsymbol{y}}^*$ to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ problem with supply $\tilde{\boldsymbol{s}}_{\boldsymbol{v}} := \tilde{\boldsymbol{s}}_{\boldsymbol{x}}$ and budget $\tilde{\boldsymbol{c}}_{\boldsymbol{v}} := \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}$. Consequently, for all supplies $\tilde{\boldsymbol{s}}_{\boldsymbol{x}}$, $\tilde{\boldsymbol{s}}_{\boldsymbol{y}}$ and budgets $\tilde{\boldsymbol{c}}_{\boldsymbol{x}}$, $\tilde{\boldsymbol{c}}_{\boldsymbol{y}}$ the following holds true

$$d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \geq d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}} + \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) = d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}),$$

where $d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}})$ is the demand label corresponding to vertex $v \in V(T)$. This implies

$$d_v(\tilde{\boldsymbol{s}}_{\boldsymbol{v}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v}}) \leq \min_{\substack{\tilde{\boldsymbol{s}}_{\boldsymbol{x}}=\tilde{\boldsymbol{s}}_{\boldsymbol{v}} \\ \tilde{\boldsymbol{s}}_{\boldsymbol{y}}=\tilde{\boldsymbol{s}}_{\boldsymbol{x}}+\boldsymbol{\beta} \\ \tilde{\boldsymbol{c}}_{\boldsymbol{v}}=\tilde{\boldsymbol{c}}_{\boldsymbol{x}}+\tilde{\boldsymbol{c}}_{\boldsymbol{y}}}} \left\{ d_x(\tilde{\boldsymbol{s}}_{\boldsymbol{x}}, \tilde{\boldsymbol{c}}_{\boldsymbol{x}}) + d_y(\tilde{\boldsymbol{s}}_{\boldsymbol{y}}, \tilde{\boldsymbol{c}}_{\boldsymbol{y}}) \right\}.$$

Finally, a robust flow in SP digraph G is obtained by backtracking the steps of the DP, and considering the demand label associated to the SP tree's root r.

Lemma 6 *Let $\boldsymbol{f}$ be an optimal robust $\boldsymbol{b}$-flow in SP digraph G_r. For the cost it holds that*

$$c(\boldsymbol{f}) = \min \left\{ \hat{c} \;\middle|\; \exists\, \tilde{\boldsymbol{c}}_{\boldsymbol{r}} \in \{0, \ldots, C_r\}^{|\Lambda|} : \max_{\lambda \in \Lambda} \tilde{c}_r^{\lambda} = \hat{c} \;\wedge\; d_r(\boldsymbol{b}(o_r), \tilde{\boldsymbol{c}}_{\boldsymbol{r}}) = 0 \right\} \qquad (7)$$

with $C_r = \sum_{a \in A(G_r)} c(a) \cdot u(a)$.

Proof For all vertices $v \in V(G) \setminus \{q\}$, the flow balance constraints of the RobMCF≡ problem are ensured by constraints (3) of the rRobMCF≡ $(\tilde{\boldsymbol{s}}_{\boldsymbol{r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{r}})$ problem with $\tilde{\boldsymbol{s}}_{\boldsymbol{r}} = \boldsymbol{b}(o_r)$. The consistent flow and capacity constraints as well as the integer conditions of the RobMCF≡ problem are one to one included in the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{r}})$ problem by constraints (4), (5) and (6), respectively. Accordingly, every feasible solution to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{r}})$ problem is also a feasible solution to the RobMCF≡ problem.

However, the rRobMCF≡ $(\tilde{\boldsymbol{s}}_{\boldsymbol{r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{r}})$ problem contains one additional set of constraints, namely constraints (2). Constraints (2) control whether the cost of a flow is equal to the budget. For this reason, we look for a budget $\tilde{\boldsymbol{c}}_{\boldsymbol{r}} \in \{0, \ldots, C_r\}^{|\Lambda|}$ for which a feasible solution to the rRobMCF≡$(\tilde{\boldsymbol{s}}_{\boldsymbol{r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{r}})$ problem exists, i.e., for which $d_r(\boldsymbol{b}(o_r), \tilde{\boldsymbol{c}}_{\boldsymbol{r}}) = 0$ holds true. This solution corresponds to a robust $\boldsymbol{b}$-flow $\boldsymbol{f}$ with cost $c(\boldsymbol{f}) = \max_{\lambda \in \Lambda} c(f^{\lambda}) = \max_{\lambda \in \Lambda} \tilde{c}_r^{\lambda}$. Therefore, we are interested in the minimum maximum budget needed among all scenarios $\lambda \in \Lambda$ which we obtain by expression (7).

After all, we analyze the runtime of the DP.

Theorem 4 *Let $(G, u, c, \boldsymbol{b})$ be a RobMCF≡ instance where G is an SP digraph with origin o. Using the DP described, the RobMCF≡ problem can be solved in $\mathcal{O}(|A(G)|(U+1)^{2|\Lambda|}(C+1)^{2|\Lambda|})$ time where $U := \sum_{a=(o,v) \in A(G)} u(a)$ and $C := \sum_{a \in A(G)} c(a) \cdot u(a)$ holds.*

Proof The correctness of the algorithm follows from Lemmas 3–6. Considering the runtime, first of all, we mention that the representation of an SP digraph G by its SP tree T can be computed in $\mathcal{O}(|A(G)|)$ (Valdes et al. 1979). At every SP tree's vertex $v \in V(T)$ demand labels for all supplies $\tilde{\boldsymbol{s}}_{\boldsymbol{v}}$ and budgets $\tilde{\boldsymbol{c}}_{\boldsymbol{v}}$ need to be calculated where the number of combinations is limited by $(U+1)^{|\Lambda|} \cdot (C+1)^{|\Lambda|}$. As SP tree T of SP digraph G has exactly $|V(T)| = 2|A(G)|-1$ vertices, we have to compute $\mathcal{O}(2|A(G)| \cdot (U+1)^{|\Lambda|} \cdot (C+1)^{|\Lambda|})$ demand labels. It remains to bound the complexity for computing the demand labels. If $v \in V(T)$ is an L-vertex, computing the corresponding demand labels is clearly in $\mathcal{O}(1)$. If $v \in V(T)$ is an S- or P-vertex, we need to compute the minimum of $(U+1)^{|\Lambda|}(C+1)^{|\Lambda|}$ sums which is in $\mathcal{O}((U+1)^{|\Lambda|}(C+1)^{|\Lambda|})$. In total, we obtain a runtime of $\mathcal{O}(2|A(G)| \cdot (U+1)^{2|\Lambda|} \cdot (C+1)^{2|\Lambda|})$.

By reason of Theorem 4, the pseudo-polynomial runtime of the DP follows if the number of scenarios is constant. Together with the result of Theorem 3 we obtain the following corollary.

Corollary 5 *The decision version of the RobMCF≡ problem on networks based on SP digraphs with multiple sources and multiple sinks is weakly $\mathcal{NP}$-complete, even if only two scenarios and unit arc capacities are considered on pearl digraphs. If the number of scenarios is not part of the input, the RobMCF≡ problem can be solved by the presented DP in pseudo-polynomial time.*

5.2 Special case of unique source and unique sink networks

In this section, we provide a polynomial time algorithm for the special case of networks based on SP digraphs with a unique source and a unique sink. The core idea of the algorithm is based on the algorithm of Bein et al. (1985) which iteratively sends flow along shortest paths to solve the MCF problem. Before we propose a generalized algorithm for the RobMCF≡ problem, we investigate properties of an optimal robust flow in the networks considered. In particular, we study the cost and show that we can restrict the statement of Lemma 2.

We start with introducing the notations and definitions needed. Let us consider a RobMCF≡ instance $(G, u, c, \boldsymbol{b})$ where G is an SP digraph with origin o and target q. As SP digraphs are acyclic, we assume without loss of generality that the unique source complies with origin o and the unique sink complies with target q. Due to Lemma 2, we limit our efforts to a set of two scenarios, i.e., $\Lambda = \{1, 2\}$. For convenience, we introduce a demand vector $\boldsymbol{d} = (d^1, d^2)$, consisting of the number of flow units that, according to the balances $\boldsymbol{b}$, are supplied from the unique source and demanded by the unique sink, i.e., $d^1 := b^1(o) = -b^1(q)$ and $d^2 := b^2(o) = -b^2(q)$. Without loss of generality, let d^1 and d^2 be given such that $d^1 \leq d^2$ holds true. Further, let $H \subseteq G$ be an SP subgraph with origin o_H. We denote the flow value of a given flow f^λ, $\lambda \in \Lambda$ entering subgraph H by $\delta(f^\lambda_{|H})$ such that the following holds

$$\delta(f^\lambda_{|H}) := \sum_{a=(o_H,w)\in A(H)} f^\lambda(a).$$

Using these notations and definitions we aim at investigating the cost of an optimal robust flow. In contrast to networks based on acyclic digraphs with a unique source and a unique sink, see Example 2, for the special case considered in this section it is sufficient to concentrate on the cost of the last scenario flow. Before we prove this statement, we need the following two auxiliary lemmas.

Lemma 7 *Let G be an SP digraph which is composed by subgraphs G_1 and G_2, and let $(G, u, c, \boldsymbol{b})$ be a corresponding RobMCF≡ instance. There exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ for which $\delta(f^2_{|G_1}) \geq \delta(f^1_{|G_1})$ and $\delta(f^2_{|G_2}) \geq \delta(f^1_{|G_2})$ hold true.*

By reason of the consistent flow constraints, the statement is not apparent. Due to the length, the proof is moved to Appendix B.

Lemma 8 *Let $(G, u, c, \boldsymbol{b})$ be a RobMCF≡ instance where G is an SP digraph. There exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ such that $f^2(a) \geq f^1(a)$ holds true for all arcs $a \in A(G)$.*

Proof Let T be the SP tree of SP digraph G. As we consider a digraph with a unique source and a unique sink, the statement of Lemma 7 can be recursively transferred to subgraphs $G_v \subseteq G$ associated to the SP tree's vertices $v \in V(T)$, i.e., $\delta(f^2_{|G_v}) \geq \delta(f^1_{|G_v})$ for all $v \in V(T)$. Consequently, $f^2(a) \geq f^1(a)$ holds true for all arcs $a \in A(G)$.

Note that, in general, the statement of Lemma 8 is not true for acyclic digraphs as Example 2 shows. We are now able to prove the following crucial lemma regarding the cost of a robust flow.

Lemma 9 *Let $(G, u, c, \boldsymbol{b})$ be a RobMCF≡ instance where G is an SP digraph. There exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ whose cost is determined by the cost of the last scenario flow, i.e., $c(\boldsymbol{f}) = \max\{c(f^1), c(f^2)\} = c(f^2)$.*

Proof By Lemma 8, there exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ such that $f^2(a) \geq f^1(a)$ holds for all arcs $a \in A(G)$. The scenario flows cause the following cost

$$c(f^1) = \sum_{a \in A} c(a) \cdot f^1(a) \leq \sum_{a \in A} c(a) \cdot f^2(a) = c(f^2),$$

from which the statement immediately follows.

By reason of Lemma 9, we concentrate on the last scenario in the following. Firstly, we note that a last scenario flow needs to send demand $d^2 - d^1$ in subgraph $G - A^{\text{fix}}$, and we refer to this demand as *excess demand*. Before we present a further useful property of the last scenario flow regarding its excess demand and a shortest path in subgraph $G - A^{\text{fix}}$, we need the following auxiliary lemma.

Lemma 10 *Let G be a series composition of SP digraphs G_1 and G_2, and let $\mathcal{I}$ be a corresponding RobMCF≡ instance. Then, let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the RobMCF≡ instances which are obtained by restricting instance $\mathcal{I}$ to subgraphs G_1 and G_2, respectively. A solution $\boldsymbol{f}$ to instance $\mathcal{I}$ is optimal if and only if the solutions $\boldsymbol{f}_{|G_1}$ and $\boldsymbol{f}_{|G_2}$, which can be obtained by restricting $\boldsymbol{f}$ to subgraphs G_1 and G_2, are optimal to instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively.*

A proof can be found in Appendix B. Example 4 in Appendix B shows that the SP property of the digraph is necessary for the truthfulness of Lemma 10. Using Lemma 10, we formulate a useful property for an existing optimal robust flow.

Lemma 11 *Let $G = (V, A = A^{\text{fix}} \cup A^{\text{free}})$ be an SP digraph with origin o and target q. Further, let $\mathcal{I} = (G, u, c, \boldsymbol{b})$ be a corresponding RobMCF≡ instance with demand $\boldsymbol{d} = (d^1, d^2)$, $d^2 \geq d^1$. With respect to cost c, let p be a shortest (o, q)-path in subgraph $G - A^{\text{fix}}$ with its bottleneck value $u_p = \min_{a \in A(p)} u(a)$. There exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ for which the following holds true*

$$f^2(a) \geq \min\{u_p, d^2 - d^1\} \text{ for all } a \in A(p). \tag{8}$$

Proof We prove the correctness of the statement by induction on the number of the digraph's arcs $m := |A|$. For the beginning, if we consider a digraph consisting of one arc only, the statement is readily apparent. In the next step, we prove the statement for a digraph with $m + 1$ arcs, providing that the statement holds true for all digraphs consisting of at most m arcs. For this purpose, we distinguish between two cases.

Firstly, we assume that G is a series composition of SP digraphs G_1 and G_2. Therefore, the origin of digraph G_1 and the target of digraph G_2 are contracted to one vertex that we denote by w. Due to the composition of digraph G, a shortest (o, q)-path p in subgraph $G - A^{\text{fix}}$ is composed of a shortest (o, w)-path p_1 in subgraph $G_1 - A^{\text{fix}}$, and a shortest (w, q)-path p_2 in subgraph $G_2 - A^{\text{fix}}$, see Fig. 5a. Considering subgraphs G_1 and G_2 separately, we obtain the RobMCF≡ instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. By induction hypothesis there exist an optimal robust flow $\boldsymbol{f}_{|G_1} = (f^1_{|G_1}, f^2_{|G_1})$ in subgraph G_1 and an optimal robust flow $\boldsymbol{f}_{|G_2} = (f^1_{|G_2}, f^2_{|G_2})$ in subgraph G_2 satisfying

$$f^2_{|G_1}(a) \geq \min\{u_{p_1}, d^2 - d^1\} \text{ for all arcs } a \in A(p_1),$$
$$f^2_{|G_2}(a) \geq \min\{u_{p_2}, d^2 - d^1\} \text{ for all arcs } a \in A(p_2).$$

By Lemma 10, the composed flow $\boldsymbol{f} = (f^1, f^2)$ with $f^1 := f^1_{|G_1} + f^1_{|G_2}$ and $f^2 := f^2_{|G_1} + f^2_{|G_2}$ is an optimal robust $\boldsymbol{b}$-flow in digraph G where the desired property is still satisfied.

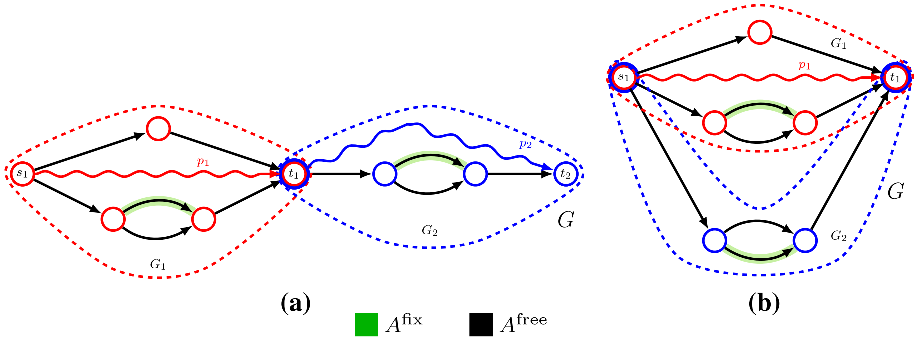

Fig. 5 Shortest path p in subgraph $G - A^{\text{fix}}$ where digraph G is a series **a** or parallel **b** composition

Secondly, we assume that G is a parallel composition of SP digraphs G_1 and G_2. Without loss of generality, let the shortest (o, q)-path p be contained in subgraph $G_1 - A^{\text{fix}}$, see Fig. 5b. Further, let $\boldsymbol{f} = (f^1, f^2)$ be an optimal robust $\boldsymbol{b}$-flow which sends demand $\boldsymbol{d} = (d^1, d^2)$ through digraph G that satisfies without loss of generality the property of Lemma 7, i.e., $\delta(f^2_{|G_i}) - \delta(f^1_{|G_i}) \geq 0$ for $G_i, i \in \{1, 2\}$. If the optimal flow $\boldsymbol{f}$ does not satisfy the property (8), applying the following procedure leads to the desired result. We consider the subgraphs G_1 and G_2 separately, resulting in RobMCF≡ instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. To define how much demand $\boldsymbol{d}_{|G_i}$ is supposed to be sent through subgraph G_i of instance $\mathcal{I}_i$, we exploit the partition of demand $\boldsymbol{d}$ of the optimal flow $\boldsymbol{f}$, i.e., $\boldsymbol{d}_{|G_i} = (d^1_{|G_i}, d^2_{|G_i}) := (\delta(f^1_{|G_i}), \delta(f^2_{|G_i}))$ for $i \in \{1, 2\}$. Considering subgraph G_1, by induction hypothesis there exists an optimal robust flow $\tilde{\boldsymbol{f}} = (\tilde{f}^1, \tilde{f}^2)$ which sends demand $\boldsymbol{d}_{|G_1} = (d^1_{|G_1}, d^2_{|G_1})$ and satisfies

$$\tilde{f}^2(a) \geq \min\{u_p, d^2_{|G_1} - d^1_{|G_1}\} \text{ for all arcs } a \in A(p).$$

Further, let an optimal robust flow $\hat{\boldsymbol{f}}$ be given which sends demand $\boldsymbol{d}_{|G_2}$ through subgraph G_2. By composing flows $\tilde{\boldsymbol{f}}$ and $\hat{\boldsymbol{f}}$, we obtain a robust $\boldsymbol{b}$-flow $\overline{\boldsymbol{f}} = (\overline{f}^1, \overline{f}^2)$ with scenario flows $\overline{f}^1 := \tilde{f}^1 + \hat{f}^1$ and $\overline{f}^2 := \tilde{f}^2 + \hat{f}^2$ for instance $\mathcal{I}$. Flow $\overline{\boldsymbol{f}}$ is optimal as there exists an optimal robust flow with the same partition $\boldsymbol{d}_{|G_1}$ and $\boldsymbol{d}_{|G_2}$ of demand $\boldsymbol{d}$ between subgraphs G_1 and G_2, and as flows $\tilde{\boldsymbol{f}}$ and $\hat{\boldsymbol{f}}$ are optimal themselves. It remains to prove that $\overline{f}^2(a) \geq \min\{u_p, d^2 - d^1\}$ holds for all arcs $a \in A(p)$. We distinguish between the following two cases.

Firstly, we consider the case where $d^2_{|G_1} - d^1_{|G_1} \geq \min\{u_p, d^2 - d^1\}$ holds true. As $\overline{f}^2(a) = \tilde{f}^2(a)$ holds for all arcs $a \in A(G_1)$ by construction, the desired property results immediately for all arcs $a \in A(p) \subseteq A(G_1)$ as shown by the following

$$\begin{aligned} \overline{f}^2(a) = \tilde{f}^2(a) &\geq \min\{u_p, d^2_{|G_1} - d^1_{|G_1}\} \\ &\geq \min\{u_p, \min\{u_p, d^2 - d^1\}\} \\ &= \min\{u_p, d^2 - d^1\}. \end{aligned}$$

Secondly, we consider the case where $d^2_{|G_1} - d^1_{|G_1} < \min\{u_p, d^2 - d^1\}$ holds true. Assume $\overline{f}^2(a) < \min\{u_p, d^2 - d^1\}$ is true for one arc $a \in A(p) \subseteq A(G_1)$. We redirect the last scenario flow $\overline{f}^2$ of robust flow $\overline{\boldsymbol{f}}$ such that demand of $\min\{u_p, d^2 - d^1\}$ is sent along the shortest path p in subgraph $G - A^{\text{fix}}$. As demand $d^2 - d^1$ needs to be sent in subgraph $G - A^{\text{fix}}$

in any case, and $A(p) \subseteq A^{\text{free}}$ holds, the resulting robust flow is still feasible, satisfies the desired property (8), and its cost is not increased.

Based on the derived knowledge by the presented lemmas, we can finally present an algorithm that solves the RobMCF≡ problem on networks based on SP digraphs with a unique source and a unique sink.

Algorithm 5.1

```
Input:   SP digraph $G = (V, A = A^{\text{fix}} \cup A^{\text{free}})$, instance $\mathcal{I} = (G, u, c, \boldsymbol{b})$, demand $\boldsymbol{d}$
Output:  Robust minimum cost $\boldsymbol{b}$-flow $\boldsymbol{f}$
Method:
    1: Compute a minimum cost flow $f'$ that sends demand $d^2 - d^1$ in subgraph $G - A^{\text{fix}}$ with respect
       to capacity $u$ and cost $c$
    2: Let $u'$ be the capacity which results from reducing the capacity $u$ of all arcs of digraph $G$ that are
       used by flow $f'$. By means of the Greedy Algorithm of Bein et al. (1985) compute a minimum
       cost flow $f''$ that sends demand $d^1$ in digraph $G$ with respect to capacity $u'$ and cost $c$, i.e., flow
       is sent along shortest paths that still have positive bottleneck values
    3: Set $f^1 := f''$ and $f^2 := f' + f''$
    4: return $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$
```

Basically, Algorithm 5.1 computes a flow by sending the excess demand in subgraph $G - A^{\text{fix}}$ first, and subsequently, by sending the demand through digraph G which is sent in both scenarios. Composing the computed flows to a robust flow leads to an optimal solution obtained in polynomial time as the following theorem shows.

Theorem 5 *Let G be an SP digraph, and let $\mathcal{I} = (G, u, c, \boldsymbol{b})$ be a corresponding RobMCF≡ instance with demand $\boldsymbol{d} = (d^1, d^2)$, $d^2 \geq d^1$. Algorithm 5.1 computes an optimal robust $\boldsymbol{b}$-flow for demand $\boldsymbol{d}$ in polynomial time.*

Proof We prove the statement by induction on the excess demand, i.e., $k := d^2 - d^1 \geq 0$. For the beginning, we consider the case where $k = 0$ holds. As the excess demand is zero, the same amount of flow needs to be sent in both scenarios. Thus, sending the excess demand in subgraph $G - A^{\text{fix}}$ in step 1 is omitted. In step 2, a minimum cost flow that sends demand d^1 through digraph G is computed by the Greedy Algorithm of Bein et al. (1985). A feasible robust flow results whose scenario flows are equal. The robust flow is optimal by the correctness of the Greedy Algorithm of Bein et al.

For the induction step, let $\tilde{\boldsymbol{f}} = (\tilde{f}^1, \tilde{f}^2)$ be an optimal robust $\boldsymbol{b}$-flow for instance $\mathcal{I}$ which satisfies without loss of generality the properties of Lemmas 7–11, i.e., in particular, $\tilde{f}^2(a) \geq \overline{u} := \min\{u_p, k+1\}$ for $a \in A(p)$. We consider the RobMCF≡ instance $\widehat{\mathcal{I}} := (G, \hat{u}, c, \hat{\boldsymbol{b}})$ with the adjusted capacity $\hat{u}$ and balances $\hat{\boldsymbol{b}} := (b^1, \hat{b}^2)$. Capacity $\hat{u}$ is obtained by reducing the capacity u of all arcs of path p by $\overline{u}$, and accordingly updating the last scenario balances b^2 of the source and sink results in the new balances $\hat{\boldsymbol{b}} := (b^1, \hat{b}^2)$. Further, we obtain the new demand $\hat{\boldsymbol{d}} = (d^1, \hat{d}^2)$ with $\hat{d}^2 := d^2 - \overline{u}$. As the excess demand is less or equal to k in instance $\widehat{\mathcal{I}}$, by induction hypothesis Algorithm 5.1 computes an optimal robust $\hat{\boldsymbol{b}}$-flow $\hat{\boldsymbol{f}} = (\hat{f}^1, \hat{f}^2)$ that sends demand $\hat{\boldsymbol{d}} = (d^1, \hat{d}^2)$. We note that robust flow $\hat{\boldsymbol{f}}$ also satisfies the properties of Lemmas 7–11. In summary, we obtain that $\hat{f}^2$ is a flow sending demand $\hat{d}^2 = d^2 - \overline{u}$ for instance $\widehat{\mathcal{I}}$, and by assumption, $\tilde{f}^2$ is an optimal last scenario flow sending demand d^2 for instance $\mathcal{I}$. Furthermore, by assumption flow $\tilde{f}^2$ sends $\overline{u}$ demand along the shortest path p in subgraph $G - A^{\text{fix}}$. Overall, we obtain for the cost of flow $\hat{f}^2$ the following upper bound

$$c(\hat{f}^2) \leq c(\tilde{f}^2) - \overline{u} \cdot c(p).$$

By reformulating, we obtain

$$c(\hat{f}^2) + \overline{u} \cdot c(p) \leq c(\tilde{f}^2),$$

and together with the condition that the cost of both flows are determined by the last scenario flows, the following holds true

$$c(\hat{\boldsymbol{f}}) + \overline{u} \cdot c(p) \leq c(\tilde{\boldsymbol{f}}).$$

Consequently, flow $\boldsymbol{f} = (f^1, f^2)$ with scenario flows $f^1 := \hat{f}^1$ and $f^2 := \hat{f}^2 + \overline{f}$ is an optimal robust $\boldsymbol{b}$-flow sending demand $\boldsymbol{d}$ where flow $\overline{f}$ is defined by $\overline{f}(a) := \overline{u}$ for all arcs $a \in A(p)$. Moreover, flow $\boldsymbol{f} = (f^1, f^2)$ complies with the flow computed by the Algorithm 5.1 for instance $\mathcal{I}$.

Finally, considering the algorithm's runtime, we compute a minimum cost flow that sends demand $d^2 - d^1$ by the Minimum Mean Cycle-Cancel Algorithm in $\mathcal{O}(|A|^3|V|^2 \log |V|)$ time (Korte et al. 2012). Subsequently, we compute a flow that sends demand d^1 by the Greedy Algorithm of Bein et al. (1985) in $\mathcal{O}(|A| \cdot |V| + |A| \log |A|)$ time. In total, computing a robust minimum cost $\boldsymbol{b}$-flow takes $\mathcal{O}(|A|^3|V|^2 \log |V| + |A| \cdot |V| + |A| \log |A|)$ time.

Note that we cannot use the Greedy Algorithm of Bein et al. (1985) in the first step of Algorithm 5.1 as $G - A^{\text{fix}}$ might not be an SP digraph.

6 Conclusion

In this paper, we introduced the RobMCF≡ problem which is an extension of the MCF problem considering equal flow requirements and demand uncertainty. We presented structural results which differentiate from well known results of the MCF problem. In particular, we showed that Dantzig and Fulkerson's Integral Flow Theorem (Korte et al. 2012) does not hold anymore. Furthermore, we proved that finding a feasible solution to the RobMCF≡ problem is strongly $\mathcal{NP}$-complete on acyclic digraphs, even if a network with a unique source, a unique sink, and unit arc capacities are considered for two scenarios only. On SP digraphs, we proved that the decision version of the RobMCF≡ problem is weakly $\mathcal{NP}$-complete. For the special case in which the number of scenarios is not part of the input, we proposed a pseudo-polynomial DP. For the special case of networks based on SP digraphs with a unique source and a unique sink, we provided an algorithm running in polynomial time.

For future work, we will study the complexity of the RobMCF≡ problem on SP digraphs if the number of scenarios is part of the input. Furthermore, we will study the RobMCF≡ problem for further graph classes as digraphs with bounded treewidth.

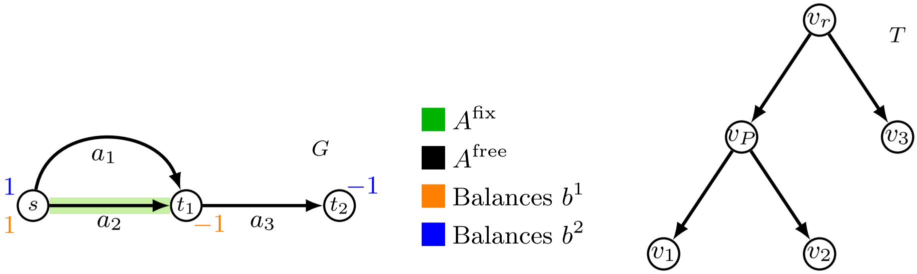

Fig. 6 SP digraph G with corresponding SP tree T for an exemplary application of the DP

Acknowledgements This study was supported by Freigeist-Fellowship of Volkswagen Stiftung.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

A Appendix-application of the dynamic program

Example 3 For a set of two scenarios $\Lambda = \{1, 2\}$, let a network $(G, u, c, \boldsymbol{b})$ with capacity $u \equiv 1$ and cost $c \equiv 0$ be given where SP digraph G, the non-zero balances $\boldsymbol{b}$, and the corresponding SP tree T are visualized in Fig. 6. For the DP, we start with initializing the leaves $v \in V(T)$. Therefore, let v_i denote the leaf which corresponds to arc $a_i \in A(G)$ for $i \in \{1, 2, 3\}$. The supply and budget are given by $\tilde{s}^{\lambda}_{v_i} \in \{0, 1\}$ and $\tilde{c}^{\lambda}_{v_i} \in \{0\}$ for $i \in \{1, 2, 3\}$, respectively. For vertices v_1 and v_3, i.e., for $i \in \{1, 3\}$, the following demand labels are equal to zero.

$$d_{v_i}(\tilde{\boldsymbol{s}}_{\boldsymbol{v_i}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v_i}}) = d_{v_i}((0,0),(0,0)) = d_{v_i}((0,1),(0,0)) = d_{v_i}((1,0),(0,0)) = d_{v_i}((1,1),(0,0)) = 0.$$

For vertex v_2, we obtain

$$d_{v_2}(\tilde{\boldsymbol{s}}_{\boldsymbol{v_2}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v_2}}) = d_{v_2}((0,0),(0,0)) = d_{v_2}((1,1),(0,0)) = 0.$$

Following the bottom-up procedure in the SP tree, we consider the P-vertex $v_P \in V(T)$ next. The supply and budget are given by $\tilde{s}^{\lambda}_{v_P} \in \{0, 1, 2\}$ and $\tilde{c}^{\lambda}_{v_P} \in \{0\}$, respectively. A computation of the demand labels of the child vertices $v_1, v_2 \in V(T)$ results in the following demand labels which are equal to zero.

$$\begin{aligned} d_{v_P}(\tilde{\boldsymbol{s}}_{\boldsymbol{v_P}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v_P}}) &= d_{v_P}((0,0),(0,0)) = d_{v_P}((0,1),(0,0)) = d_{v_P}((1,0),(0,0)) = d_{v_P}((1,1),(0,0)) \\ &= d_{v_P}((1,2),(0,0)) = d_{v_P}((2,1),(0,0)) = d_{v_P}((2,2),(0,0)) = 0. \end{aligned}$$

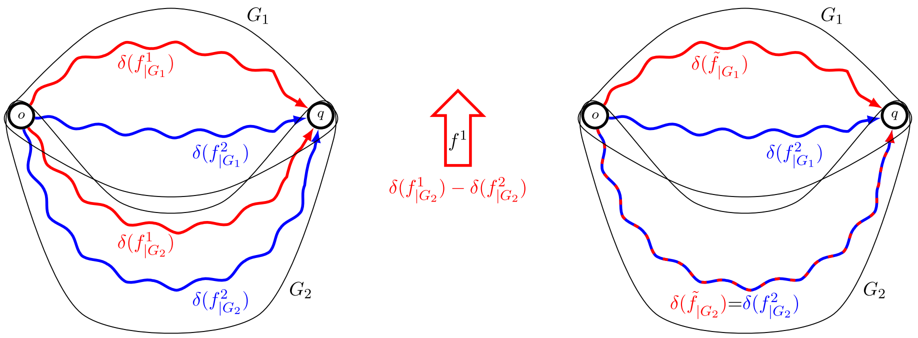

Fig. 7 Shifting $\delta(f^1_{|G_2}) - \delta(f^2_{|G_2})$ demand of the first scenario flow f^1 from G_2 to G_1 leads to a new flow $\tilde{f}$ that satisfies the desired property

In the next step, we consider the S-vertex $v_r \in V(T)$ which builds the SP tree's root r. The supply and budget are given by $\tilde{s}^{\lambda}_{v_r} \in \{0, 1, 2\}$ and $\tilde{c}^{\lambda}_{v_r} \in \{0\}$, respectively. A computation of the demand labels of the child vertices $v_P, v_3 \in V(T)$ results in the following demand labels which are equal to zero.

$$\begin{aligned} d_{v_r}(\tilde{\boldsymbol{s}}_{\boldsymbol{v_r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v_r}}) &= d_{v_r}((1,0),(0,0)) = d_{v_r}((1,1),(0,0)) = d_{v_r}((1,2),(0,0)) = d_{v_r}((2,1),(0,0)) \\ &= d_{v_r}((2,2),(0,0)) = 0. \end{aligned}$$

To determine the cost of an optimal robust $\boldsymbol{b}$-flow in SP digraph G, we exploit Lemma 6. Therefore, we need to consider all demand labels $d_{v_r}(\tilde{\boldsymbol{s}}_{\boldsymbol{v_r}}, \tilde{\boldsymbol{c}}_{\boldsymbol{v_r}})$ with $\tilde{\boldsymbol{s}}_{\boldsymbol{v_r}} = \boldsymbol{b}(s) = (1, 1)$ and choose the one with the minimum maximum budget among all scenarios $\lambda \in \Lambda$, i.e., $d_{v_r}((1,1),(0,0))$. As a result, the cost of an optimal robust $\boldsymbol{b}$-flow in SP digraph G is zero and it can be recursively constructed by the demand labels.

B Appendix-omitted proofs

Lemma 7 *Let G be an SP digraph which is composed by subgraphs G_1 and G_2, and let $(G, u, c, \boldsymbol{b})$ be a corresponding RobMCF≡ instance. There exists an optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f} = (f^1, f^2)$ for which $\delta(f^2_{|G_1}) \geq \delta(f^1_{|G_1})$ and $\delta(f^2_{|G_2}) \geq \delta(f^1_{|G_2})$ hold true.*

Proof Let $\boldsymbol{f} = (f^1, f^2)$ be an optimal robust $\boldsymbol{b}$-flow which sends demand $\boldsymbol{d} = (d^1, d^2)$ through digraph G. We distinguish whether digraph G is a series or parallel composition of subgraphs G_1 and G_2. For the case that digraph G is serially composed the validity of the statement is apparent as we consider a network with a unique source and a unique sink, and $d^2 \geq d^1$ holds. In case that digraph G is parallelly composed the following is true. If $d^2 = d^1$ holds, the statement is again apparent. Otherwise, if $d^2 > d^1$ holds, $\delta(f^2_{|G_i}) > \delta(f^1_{|G_i})$ also holds true for at least one of the subgraphs G_i, $i \in \{1, 2\}$. Without loss of generality, let G_1 be the subgraph for which $\delta(f^2_{|G_1}) > \delta(f^1_{|G_1})$ holds true, and in return, assume that $\delta(f^2_{|G_2}) < \delta(f^1_{|G_2})$ holds. In the following, we provide a procedure by which we redirect a proportion of the scenario flows f^1 or f^2 such that the desired property holds.

In the first step, we define two new scenario flows $\tilde{f}$ and $\hat{f}$ that send demand $\delta(\tilde{f}_{|G}) = d^1$ and $\delta(\hat{f}_{|G}) = d^2$ through digraph G, respectively. Flow $\tilde{f}$ corresponds to a first scenario flow which is obtained by redirecting a proportion of flow f^1 from subgraph G_2 to subgraph G_1

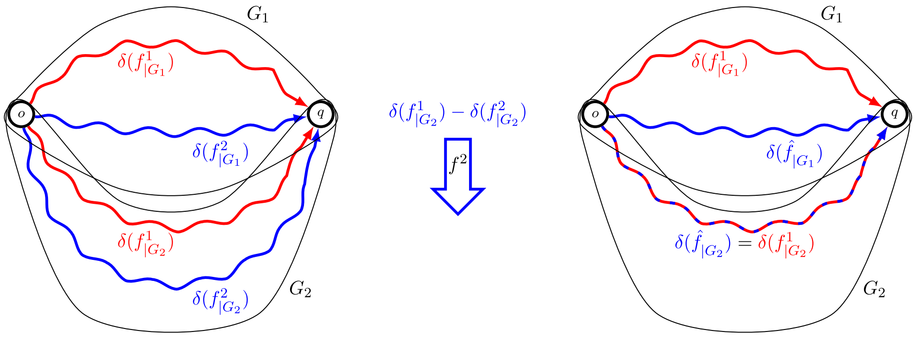

Fig. 8 Shifting $\delta(f^1_{|G_2}) - \delta(f^2_{|G_2})$ demand of the last scenario flow f^2 from G_1 to G_2 leads to a new flow $\hat{f}$ that satisfies the desired property

such that $\delta(f^2_{|G_2}) \geq \delta(\tilde{f}_{|G_2})$ holds true, see Fig. 7. More precisely, flow $\tilde{f} := \tilde{f}_{|G_1} + \tilde{f}_{|G_2}$ is defined in such a way that demand $\tilde{d}_1 := \delta(\tilde{f}_{|G_1}) = \delta(f^1_{|G_1}) + (\delta(f^1_{|G_2}) - \delta(f^2_{|G_2}))$ is sent through subgraph G_1, and demand $\tilde{d}_2 := \delta(\tilde{f}_{|G_2}) = \delta(f^2_{|G_2})$ through subgraph G_2. Considering subgraph G_1, by assumption and definition it holds $\delta(f^1_{|G_1}) < \tilde{d}_1 < \delta(f^2_{|G_1})$. Following Lemma 2, we compute a robust flow $\overline{\boldsymbol{f}} = (f^1_{|G_1}, \tilde{f}_{|G_1}, f^2_{|G_1})$ sending demand $\overline{\boldsymbol{d}} = (\delta(f^1_{|G_1}), \tilde{d}_1, \delta(f^2_{|G_1}))$ through subgraph G_1 and causing cost of $c(\overline{\boldsymbol{f}}) = \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\}$. We further set $\tilde{f}_{|G_2} := f^2_{|G_2}$ such that the overall cost of flow $\tilde{f}$ is estimated as follows

$$c(\tilde{f}_{|G_1}) \leq \max\left\{c\left(f^1_{|G_1}\right), c\left(f^2_{|G_1}\right)\right\},$$
$$c(\tilde{f}_{|G_2}) = c(f^2_{|G_2}).$$

Flow $\hat{f}$ in turn corresponds to a last scenario flow which is obtained by redirecting a proportion of flow f^2 from subgraph G_1 to subgraph G_2 such that $\delta(\hat{f}_{|G_2}) \geq \delta(f^1_{|G_2})$ holds true, see Fig. 8. More precisely, we define the scenario flow $\hat{f} := \hat{f}_{|G_1} + \hat{f}_{|G_2}$ such that demand $\hat{d}_1 := \delta(\hat{f}_{|G_1}) = \delta(f^2_{|G_1}) - (\delta(f^1_{|G_2}) - \delta(f^2_{|G_2}))$ is sent through subgraph G_1, and demand $\hat{d}_2 := \delta(\hat{f}_{|G_2}) = \delta(f^1_{|G_2})$ through subgraph G_2. Considering subgraph G_1, by assumption and definition it holds $\delta(f^1_{|G_1}) < \hat{d}_1 < \delta(f^2_{|G_1})$. Following Lemma 2, we compute a robust flow $\underline{\boldsymbol{f}} = (f^1_{|G_1}, \hat{f}_{|G_1}, f^2_{|G_1})$ sending demand $\underline{\boldsymbol{d}} = (\delta(f^1_{|G_1}), \hat{d}_1, \delta(f^2_{|G_1}))$ in subgraph G_1 and causing cost of $c(\underline{\boldsymbol{f}}) = \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\}$. We further set $\hat{f}_{|G_2} := f^1_{|G_2}$ and obtain the following estimations of the cost

$$c\left(\hat{f}_{|G_1}\right) \leq \max\left\{c\left(f^1_{|G_1}\right), c\left(f^2_{|G_1}\right)\right\},$$
$$c\left(\hat{f}_{|G_2}\right) = c\left(f^1_{|G_2}\right).$$

In the next step, we construct two new robust $\boldsymbol{b}$-flows $\boldsymbol{f_a} := (f^1, \hat{f})$ and $\boldsymbol{f_b} := (\tilde{f}, f^2)$ which are obtained by redirecting the scenario flows of the optimal robust $\boldsymbol{b}$-flow $\boldsymbol{f}$. The robust flows $\boldsymbol{f_a}$ and $\boldsymbol{f_b}$ are feasible by construction of flows $\hat{f}$ and $\tilde{f}$ and each sends demand $\boldsymbol{d}$ through digraph G. If we show that $\min\{c(\boldsymbol{f_a}), c(\boldsymbol{f_b})\} \leq c(\boldsymbol{f})$ holds true, we can redirect the optimal robust flow $\boldsymbol{f}$ analogous to either robust flow $\boldsymbol{f_a}$ or $\boldsymbol{f_b}$ such that the desired property is satisfied but the cost is not changed. We distinguish whether the first or last scenario flow of the optimal robust solution $\boldsymbol{f}$ is more expensive. Firstly, we assume that the first scenario flow f^1 is more expensive than the last scenario flow f^2, i.e., $c(\boldsymbol{f}) = \max\{c(f^1), c(f^2)\} = c(f^1)$. Further, we distinguish between the following two cases.

1. Case $c(f^1_{|G_1}) \geq c(f^2_{|G_1})$
To prove the statement $\min\{c(\boldsymbol{f_a}), c(\boldsymbol{f_b})\} \leq c(\boldsymbol{f})$, it is sufficient to prove the statement $c(\boldsymbol{f_a}) \leq c(\boldsymbol{f})$. By equivalent transformation we obtain

$$\begin{aligned} & c(\boldsymbol{f_a}) \leq c(\boldsymbol{f}) \\ \Leftrightarrow \quad & \max\{c(f^1), c(\hat{f})\} \leq \max\{c(f^1), c(f^2)\} \\ \Leftrightarrow \quad & \max\{c(f^1), c(\hat{f})\} \leq c(f^1). \end{aligned}$$

Consequently, we only need to prove that $c(\hat{f}) \leq c(f^1)$ holds. Using the definition and cost estimation of flow $\hat{f}$, we can alternatively show the following

$$\max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^1_{|G_2}) \leq c(f^1_{|G_1}) + c(f^1_{|G_2}).$$

Equivalent transforming results in

$$\begin{aligned} & \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^1_{|G_2}) \leq c(f^1_{|G_1}) + c(f^1_{|G_2}) \\ \Leftrightarrow \quad & \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} \leq c(f^1_{|G_1}) \\ \Leftrightarrow \quad & c(f^1_{|G_1}) \leq c(f^1_{|G_1}), \end{aligned}$$

which is a true statement.
2. Case $c(f^1_{|G_1}) < c(f^2_{|G_1})$
To prove the statement $\min\{c(\boldsymbol{f_a}), c(\boldsymbol{f_b})\} \leq c(\boldsymbol{f})$, it is sufficient to prove the statement $c(\boldsymbol{f_b}) \leq c(\boldsymbol{f})$. For this case, the cost of the robust flow $\boldsymbol{f_b} = (\tilde{f}, f^2)$ is determined by flow f^2 as shown by the following

$$\begin{aligned} c(\tilde{f}) = c(\tilde{f}_{|G_1}) + c(\tilde{f}_{|G_2}) &\leq \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^2_{|G_2}) \\ &= c(f^2_{|G_1}) + c(f^2_{|G_2}) = c(f^2). \end{aligned}$$

Accordingly, equivalent transformation results in

$$\begin{aligned} & c(\boldsymbol{f_b}) \leq c(\boldsymbol{f}) \\ \Leftrightarrow \quad & \max\{c(\tilde{f}), c(f^2)\} \leq \max\{c(f^1), c(f^2)\} \\ \Leftrightarrow \quad & c(f^2) \leq c(f^1), \end{aligned}$$

which is a true statement for the present case.

Secondly, we assume that the last scenario flow f^2 is more expensive than the first scenario flow f^1, i.e., $c(\boldsymbol{f}) = \max\{c(f^1), c(f^2)\} = c(f^2)$. Further, we distinguish between the following two cases.

1. Case $c(f^2_{|G_1}) \geq c(f^1_{|G_1})$
To prove the statement $\min\{c(\boldsymbol{f_a}), c(\boldsymbol{f_b})\} \leq c(\boldsymbol{f})$, it is sufficient to prove the statement $c(\boldsymbol{f_b}) \leq c(\boldsymbol{f})$. By equivalent transformation we obtain

$$\begin{aligned} & c(\boldsymbol{f_b}) \leq c(\boldsymbol{f}) \\ \Leftrightarrow \quad & \max\{c(\tilde{f}), c(f^2)\} \leq \max\{c(f^1), c(f^2)\} \end{aligned}$$

$$\Leftrightarrow \qquad \max\{c(\tilde{f}), c(f^2)\} \leq c(f^2).$$

Consequently, we only need to prove that $c(\tilde{f}) \leq c(f^2)$ holds. Using the definition and cost estimation of flow $\tilde{f}$, we can alternatively show the following

$$\max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^2_{|G_2}) \leq c(f^2_{|G_1}) + c(f^2_{|G_2}).$$

Equivalent transforming results in

$$\begin{aligned} & \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^2_{|G_2}) \leq c(f^2_{|G_1}) + c(f^2_{|G_2}) \\ \Leftrightarrow \quad & \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} \leq c(f^2_{|G_1}) \\ \Leftrightarrow \quad & c(f^2_{|G_1}) \leq c(f^2_{|G_1}), \end{aligned}$$

which is a true statement.
2. Case $c(f^2_{|G_1}) < c(f^1_{|G_1})$
To prove the statement $\min\{c(\boldsymbol{f_a}), c(\boldsymbol{f_b})\} \leq c(\boldsymbol{f})$, it is sufficient to prove the statement $c(\boldsymbol{f_a}) \leq c(\boldsymbol{f})$. For this case, the cost of the robust flow $\boldsymbol{f_a} = (f^1, \hat{f})$ is determined by flow f^1 as shown by the following

$$\begin{aligned} c(\hat{f}) = c(\hat{f}_{|G_1}) + c(\hat{f}_{|G_2}) &\leq \max\{c(f^1_{|G_1}), c(f^2_{|G_1})\} + c(f^1_{|G_2}) \\ &= c(f^1_{|G_1}) + c(f^1_{|G_2}) = c(f^1). \end{aligned}$$

Accordingly, equivalent transformation results in

$$\begin{aligned} & c(\boldsymbol{f_a}) \leq c(\boldsymbol{f}) \\ \Leftrightarrow \quad & \max\left\{c(f^1), c(\hat{f})\right\} \leq \max\left\{c(f^1), c(f^2)\right\} \\ \Leftrightarrow \quad & c(f^1) \leq c(f^2), \end{aligned}$$

which is a true statement for the present case.

In summary, by redirecting the scenario flows of the optimal $\boldsymbol{b}$-flow $\boldsymbol{f}$ we obtain the desired property without changing the cost.

Lemma 10 *Let G be a series composition of SP digraphs G_1 and G_2, and let $\mathcal{I}$ be a corresponding RobMCF≡ instance. Then, let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the RobMCF≡ instances which are obtained by restricting instance $\mathcal{I}$ to subgraphs G_1 and G_2, respectively. A solution $\boldsymbol{f}$ to instance $\mathcal{I}$ is optimal if and only if the solutions $\boldsymbol{f}_{|G_1}$ and $\boldsymbol{f}_{|G_2}$, which can be obtained by restricting $\boldsymbol{f}$ to subgraphs G_1 and G_2, are optimal to instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively.*

Proof Without loss of generality, we assume that the robust flows given in this proof satisfy the property of Lemma 9. Let $\boldsymbol{f} = (f^1, f^2)$ be an optimal robust flow for instance $\mathcal{I}$. If we restrict flow $\boldsymbol{f}$ to subgraphs G_1 and G_2, feasible flows $\boldsymbol{f}_{|G_1} = (f^1_{|G_1}, f^2_{|G_1})$ and $\boldsymbol{f}_{|G_2} = (f^1_{|G_2}, f^2_{|G_2})$ result for instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. Furthermore, they still satisfy the property of Lemma 9. Assume that flow $\boldsymbol{f}_{|G_1}$ is not optimal for instance $\mathcal{I}_1$. Consequently, there exists an optimal robust flow $\tilde{\boldsymbol{f}} = (\tilde{f}^1, \tilde{f}^2)$ in subgraph G_1 with less cost, i.e.,

$$c(\tilde{\boldsymbol{f}}) = \max\left\{c(\tilde{f}^1), c(\tilde{f}^2)\right\} = c(\tilde{f}^2) < c(f^2_{|G_1}) = \max\left\{c(f^1_{|G_1}), c(f^2_{|G_1})\right\} = c(\boldsymbol{f}_{|G_1}).$$

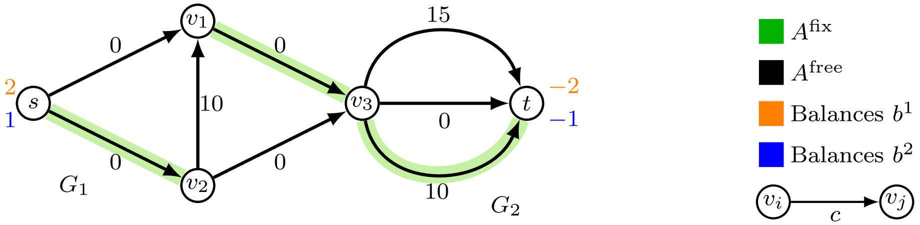

Fig. 9 An optimal solution in subgraph G_1 composed with an optimal solution in subgraph G_2 is not optimal in digraph G composed by digraphs G_1 and G_2

However, this means that the composed flows $\hat{f}^1 := \tilde{f}^1 + f^1_{|G_2}$ and $\hat{f}^2 := \tilde{f}^2 + f^2_{|G_2}$ result in a feasible robust flow $\hat{\boldsymbol{f}} = (\hat{f}^1, \hat{f}^2)$ with cost

$$\begin{aligned} c(\hat{\boldsymbol{f}}) &= \max\left\{c(\hat{f}^1), c(\hat{f}^2)\right\} \\ &= \max\left\{c(\tilde{f}^1) + c(f^1_{|G_2}), c(\tilde{f}^2) + c(f^2_{|G_2})\right\} \\ &= c(\tilde{f}^2) + c(f^2_{|G_2}) \\ &< c(f^2_{|G_1}) + c(f^2_{|G_2}) = c(f^2) = c(\boldsymbol{f}), \end{aligned}$$

which contradicts to the assumption. The optimality of flow $\boldsymbol{f}_{|G_2} = (f^1_{|G_2}, f^2_{|G_2})$ follows for instance $\mathcal{I}_2$ due to the analog argumentation.

Conversely, let $\boldsymbol{f}_{|G_1}$ and $\boldsymbol{f}_{|G_2}$ be optimal flows for instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. The composition of these flows results in a feasible robust flow $\boldsymbol{f} := \boldsymbol{f}_{|G_1} + \boldsymbol{f}_{|G_2}$ for instance $\mathcal{I}$ that causes cost of

$$\begin{aligned} c(\boldsymbol{f}) &= \max\left\{c(f^1), c(f^2)\right\} \\ &= \max\left\{c(f^1_{|G_1}) + c(f^1_{|G_2}), c(f^2_{|G_1}) + c(f^2_{|G_2})\right\} \\ &= c(f^2_{|G_1}) + c(f^2_{|G_2}) \\ &= c(f^2). \end{aligned}$$

Assume the robust flow $\boldsymbol{f}$ is not optimal which in turn means that there exists an optimal robust flow $\tilde{\boldsymbol{f}}$ with less cost, i.e. $c(\tilde{\boldsymbol{f}}) < c(\boldsymbol{f})$. As flows $\boldsymbol{f}_{|G_1}$ and $\boldsymbol{f}_{|G_2}$ are optimal for instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively, $c(f^2_{|G_i}) \leq c(\tilde{f}^2_{|G_i})$ holds true for both subgraphs $G_i, i \in \{1, 2\}$. Overall, we obtain

$$c(\boldsymbol{f}) = c(f^2) = c(f^2_{|G_1}) + c(f^2_{|G_2}) \leq c(\tilde{f}^2_{|G_1}) + c(\tilde{f}^2_{|G_2}) = c(\tilde{f}^2) = c(\tilde{\boldsymbol{f}}),$$

which is a contradiction to the assumption.

Example 4 For a set of two scenarios $\Lambda = \{1, 2\}$, let a network $(G, u, c, \boldsymbol{b})$ with capacity $u \equiv 1$ be given where digraph G, its cost c, and the non-zero balances $\boldsymbol{b}$ are visualized in Fig. 9. An optimal solution $\boldsymbol{f} = (f^1, f^2)$ to the RobMCF≡ problem can be easily established. Considering the second scenario flow f^2 first, the only option to send two flow units from source s to sink t is along paths sv_1v_3t and sv_2v_3t due to the capacity constraints. As the second scenario flow f^2 uses both fixed arcs in subgraph G_1, the first scenario flow f^1 must also send flow along these arcs. For this reason, the only option to send one flow unit from source s to sink t is along the path $sv_2v_1v_3t$. Concentrating on subgraph G_1, flow f^1

causes cost of ten while flow f^2 does not cause any cost. Since the overall aim is to construct a robust $\boldsymbol{b}$-flow with minimum cost, flow f^1 sends the flow unit via the second parallel arc of multi-arc (v_3, t) in subgraph G_2 causing zero cost. Flow f^2 also sends one flow unit via this arc, and additionally one flow unit via the first parallel arc of multi-arc (v_3, t) causing cost of 15. In total, we obtain cost of

$$c(\boldsymbol{f}) = \max\left\{c(f^1), c(f^2)\right\} = \max\{10 + 0, 0 + 15\} = 15 = c(f^2).$$

Due to the construction of digraph G, sending flow along paths from source s to sink t requires the usage of vertex v_3 which connects the subgraphs G_1 and G_2. For this reason, we consider in the next step the RobMCF≡ problem on the subgraphs G_1 and G_2 separately. Therefore, let $\mathcal{I}_1 = (G_1, u, c, \tilde{\boldsymbol{b}})$ be the RobMCF≡ instance restricted to subgraph G_1 with newly defined balances by

$$\tilde{\boldsymbol{b}}(v) = \begin{cases} \boldsymbol{b}(v) \text{ for all } v \in V(G_1) \setminus \{v_3\}, \\ \boldsymbol{b}(t) \text{ for } v = v_3. \end{cases}$$

An optimal solution $\tilde{\boldsymbol{f}} = (\tilde{f}^1, \tilde{f}^2)$ to instance $\mathcal{I}_1$ is equal to solution $\boldsymbol{f}$ restricted to subgraph G_1, and causes cost of

$$c(\tilde{\boldsymbol{f}}) = \max\{c(\tilde{f}^1), c(\tilde{f}^2)\} = \max\{10, 0\} = 10.$$

Further, let $\mathcal{I}_2 = (G_2, u, c, \hat{\boldsymbol{b}})$ be the RobMCF≡ instance restricted to subgraph G_2 with balances $\hat{\boldsymbol{b}}(v_3) = \boldsymbol{b}(s)$ and $\hat{\boldsymbol{b}}(t) = \boldsymbol{b}(t)$. An optimal solution $\hat{\boldsymbol{f}} = (\hat{f}^1, \hat{f}^2)$ to instance $\mathcal{I}_2$ is determined as follows. Both scenario flows $\hat{f}^1$ and $\hat{f}^2$ send one flow unit along the third parallel arc of multi-arc (v_3, t) while the second scenario flow $\hat{f}^2$ additionally sends one flow unit along the second parallel arc. This ends up in cost of

$$c(\hat{\boldsymbol{f}}) = \max\left\{c(\hat{f}^1), c(\hat{f}^2)\right\} = \max\{10, 10 + 0\} = 10.$$

Consequently, the optimal solution $\hat{\boldsymbol{f}}$ in subgraph G_2 causes less cost than the optimal solution $\boldsymbol{f}$ in digraph G restricted to subgraph G_2 which causes cost of 15.
Conversely, the solution, which results if optimal solutions $\tilde{\boldsymbol{f}}$ and $\hat{\boldsymbol{f}}$ to instances $\mathcal{I}_1$ and $\mathcal{I}_2$ are composed, is feasible but not optimal for instance $\mathcal{I}$.

References

Ahuja, R.K., Magnanti,T.L.,& Orlin,J.B. (1988). *Network flows*.

Ahuja, R. K., Orlin, J. B., Sechi, G. M., & Zuddas, P. (1999). Algorithms for the simple equal flow problem. *Management Science, 45*(10), 1440–1455.

Ali, A. I., Kennington, J., & Shetty, B. (1988). The equal flow problem. *European Journal of Operational Research, 36*(1), 107–115.

Altin, A., Amaldi, E., Belotti, P., & Pinar, M. Ç. (2007). Provisioning virtual private networks under traffic uncertainty. *Networks, 49*(1), 100–155.

Altin, A., Yaman, H., & Pinar, M. (2011). The robust network loading problem under hose demand uncertainty: formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing, 23*(1), 75–89.

Álvarez-Miranda, E., Cacchiani, V., Dorneth, T., Jünger, M., Liers, F., Lodi, A., Parriani, T., & Schmidt, D.R. (2012). Models and algorithms for robust network design with several traffic scenarios. In A. Ridha Mahjoub, V. Markakis, I. Milis, V.T. Paschos (Eds), *ISCO 2012, revised selected papers* (vol 7422, pp. 261–272),

Atamtürk, A., & Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research, 55*(4), 662–673.

Bein, W. W., Brucker, P., & Tamir, A. (1985). Minimum cost flow algorithms for series-parallel networks. *Discrete Applied Mathematics, 10*(2), 117–124.

Belotti, P., Capone, A., Carello, G., & Malucelli, F. (2008). Multi-layer mpls network design: the impact of statistical multiplexing. *Computer Networks, 52*(6), 1291–1307.

Berman, P., Karpinski, M., & Scott, A. (2004). *Approximation hardness of short symmetric instances of max-3sat*. Technical report

Bertsimas, D., & Sim, M. (2003). *Robust Discrete Optimization and Network Flows, 98*(1), 49–71.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53.

Cacchiani, V., Jünger, M., Liers, F., Lodi, A., & Schmidt, D. R. (2016). Single-commodity robust network design with finite and hose demand sets. *Mathematical Programming, 157*(1), 297–342.

Calvete, H. I. (2003). Network simplex algorithm for the general equal flow problem. *European Journal of Operational Research, 150*(3), 585–600.

Carraresi, P., & Gallo, G. (1984). Network models for vehicle and crew scheduling. *European Journal of Operational Research, 16*(2), 139–151.

Even, S., Itai, A., Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *16th annual symposium on foundations of computer science (sfcs 1975)* (pp. 184–193). IEEE.

Johnson, D.S., & Garey, M.R. (1979) *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman.

Korte, B., Vygen, J., Korte, B., & Vygen, J. (2012). *Combinatorial optimization* (Vol. 2). Berlin: Springer.

Koster, A. M. C. A., Kutschka, M., & Raack, C. (2013). Robust network design: formulations, valid inequalities, and computations. *Networks, 61*(2), 128–149. https://doi.org/10.1002/net.21497.

Krumke, S. O., & Thielen, C. (2011). Minimum cost flows with minimum quantities. *Information Processing Letters, 111*(11), 533–537.

LenstraJr, H. W. (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research, 8*(4), 538–548.

Manca, A., Sechi, G. M., & Zuddas, P. (2010). Water supply network optimisation using equal flow algorithms. *Water Resources Management, 24*(13), 3665–3678.

Mattia, S. (2013). The robust network loading problem with dynamic routing. *Computational Optimization and Applications, 54,* 619–643.

Meyers, C. A., & Schulz, A. S. (2009). Integer equal flows. *Operations Research Letters, 37*(4), 245–249.

Minoux, M. (1989). Networks synthesis and optimum network design problems: models, solution methods and applications. *Networks, 19*(3), 313–360.

Morrison, D. R., Sauppe, J. J., & Jacobson, S. H. (2013). A network simplex algorithm for the equal flow problem on a generalized network. *INFORMS Journal on Computing, 25*(1), 2–12.

Ohst, J.P. (2016). On the construction of optimal paths from flows and the analysis of evacuation scenarios.

Pferschy, U., & Schauer, J. (2013). The maximum flow problem with disjunctive constraints. *Journal of Combinatorial Optimization, 26*(1), 109–119.

Poss, M., & Raack, C. (2013). Affine recourse for the robust network design problem: between static and dynamic routing. *Networks*, *61*(2)

Sahni, S. (1974). Computationally related problems. *SIAM Journal on Computing, 3*(4), 262–279.

Sanità, L. (2009). *Robust Network Design*. PhD thesis, Università La Sapienza, Roma.

Seedig, H.G. (2009). Network flow optimization with minimum quantities. In *Operations research proceedings 2010* pp. 295–300. Springer.

Srinathan, K., Goundan, P.R., Kumar, M.V., Nandakumar, R.,& Rangan, C.P. (2002). Theory of equal-flows in networks. In O.H. Ibarra, L. Zhang, (Eds), *Computing and combinatorics* (pp. 514–524), Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN 978-3-540-45655-1.

Valdes,J., Tarjan,R.E., & Lawler, E.L. (1979). The recognition of series parallel digraphs. In *Proceedings of the eleventh annual ACM symposium on theory of computing, STOC '79* (pp. 1–12) New York, NY, USA, ACM. https://doi.org/10.1145/800135.804393.

Valdes, J., Tarjan, R. E., & Lawler, E. L. (1982). The recognition of series parallel digraphs. *SIAM Journal on Computing, 11*(2), 298–313.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.