Robust transshipment problem under consistent flow constraints

Christina Büsing1 | Arie M. C. A. Koster2 | Sabrina Schmitz1

1Combinatorial Optimization, RWTH Aachen University, Aachen, Germany
2Discrete Optimization, RWTH Aachen University, Aachen, Germany

Correspondence
Sabrina Schmitz, Combinatorial Optimization, RWTH Aachen University, Aachen, 52062, Germany.
Email: schmitz@combi.rwth-aachen.de

Funding information
Deutsche Forschungsgemeinschaft;
Freigeist-Fellowship Volkswagen Stiftung.

Abstract
In this article, we study robust transshipment under consistent flow constraints. We consider demand uncertainty represented by a finite set of scenarios and characterize a subset of arcs as so-called fixed arcs. In each scenario, we require an integral flow that satisfies the respective flow balance constraints. In addition, on each fixed arc, we require equal flow for all scenarios. The objective is to minimize the maximum cost occurring among all scenarios. We show that the problem is strongly \( \mathbb{NP} \)-complete on acyclic digraphs by a reduction from the \((3, B2)\)-SAT problem. Furthermore, we prove that the problem is weakly \( \mathbb{NP} \)-complete on series-parallel digraphs by a reduction from a special case of the Partition problem. If in addition the number of scenarios is constant, we observe the pseudo-polynomial-time solvability of the problem. We provide polynomial-time algorithms for three special cases on series-parallel digraphs. Finally, we present a polynomial-time algorithm for pearl digraphs.

KEYWORDS
demand uncertainty, equal flow problem, minimum cost flow, robust flows, series-parallel digraphs, transshipment problem

1 INTRODUCTION

In this article, we consider the robust transshipment problem under consistent flow constraints (ROBT≡). The problem is motivated by long-term decisions on transshipment that have to be made despite uncertainties in demand. For instance, in logistic applications the transshipment is often agreed in advance by long-term contracts with subcontractors. Once an agreement is signed, the transshipment needs to be performed even if demand fluctuates. A solution to the ROBT≡ problem facilitates cost-efficient decision-making which is robust against demand uncertainty.

As in the transshipment problem [15], we consider an uncapacitated network in the ROBT≡ problem. To represent demand uncertainty, we consider vertex balances for a finite number of scenarios. Furthermore, we characterize a subset of arcs as so-called fixed arcs. In each scenario, we require an integral flow that satisfies the respective flow balance constraints. The flow on a fixed arc represents the transshipment by subcontractors. For this reason, on each fixed arc, we require equal flow for all scenarios to determine a transshipment that is robust against demand uncertainty. The objective is to minimize the maximum cost that may occur among all scenarios. We note that the integral requirement for the flow is necessary, even though the balances of all scenarios are integral, as Dantzig and Fulkerson’s Integral Flow Theorem [15] does not hold for the ROBT≡ problem. A formal problem definition is provided in Section 2.

The ROBT≡ problem is the uncapacitated version of the robust minimum cost flow problem under consistent flow constraints (ROBMCF≡), introduced in our previous work [9]. The uncapacitated version is of interest, even if logistic applications might have a limited transshipment capacity, as the limitations are often sufficiently large. Since the complexity results for the
The ROBMCF≡ problem rely on the arc capacities of the network, the question arises whether the problem is solvable in polynomial time if the capacity restrictions are neglected. This is the case, for example, for the integral multi-commodity flow problem, which is \(\mathcal{NP}\)-hard in general but solvable in polynomial time for uncapacitated networks [15]. We note that the ROBMCF≡ and ROBT≡ problems are not equivalent as a transformation from the ROBMCF≡ problem to the ROBT≡ problem changes not only the digraph structure but also the start and target locations of the transshipment as well as the amount to be shipped.

The main contribution of this article is the analysis of the ROBT≡ problem on series-parallel (SP) digraphs summarized as follows. First, we prove that the decision version of the ROBT≡ problem is weakly \(\mathcal{NP}\)-complete, even if only two scenarios are considered that have the same unique source and single but different sinks. Second, we present a polynomial-time algorithm for the special case of a unique source and so-called parallel sinks. We derive the same result for the special case of so-called parallel sources and a unique sink. Furthermore, we present that the algorithm is extendable for the special case of parallel sources and parallel sinks if there exists a path between each source and each sink. Finally, we present a polynomial-time algorithm for the special case of pearl digraphs, independent of the number of sources and sinks.

On the basis of our previous work [9] we present the following minor contribution. First, we prove that deciding whether a feasible solution to the ROBT≡ problem exists is strongly \(\mathcal{NP}\)-complete on acyclic digraphs, even if only two scenarios are considered that have the same unique source and unique sink. Second, we observe the pseudo-polynomial-time solvability on SP digraphs for the special case of a constant number of scenarios. Third, we establish the polynomial-time solvability on SP digraphs for the special case of parallel sources (sinks). In Section 4, we analyze the complexity of the ROBT≡ problem on SP digraphs in general, on SP digraphs with a unique source and a unique sink, and on SP digraphs with a unique source and parallel sinks. In Section 5, we analyze the complexity of the ROBT≡ problem on pearl digraphs. In Section 6, we conclude our results.

## 2 | PROBLEM DEFINITION AND CLASSIFICATION

### 2.1 | Definition and notations

Let digraph \(G = (V, A)\) be given with vertex set \(V\) and arc set \(A\). The set of arcs \(A\) is partitioned into two disjoint sets \(A_{\text{fix}}\) and \(A_{\text{free}}\), termed fixed and free arcs, respectively. If not explicitly defined, we specify the sets of vertices, arcs, fixed arcs, and free arcs of a digraph \(G\) by \(V(G), A(G), A_{\text{fix}}(G)\), and \(A_{\text{free}}(G)\), respectively. Let arc cost \(c : A \rightarrow \mathbb{Z}_{\geq 0}\) be given. The demand uncertainty is represented by the finite set of scenarios \(\Lambda\). For every scenario \(\lambda \in \Lambda\), vertex balances \(b^\lambda : V \rightarrow \mathbb{Z}\) with \(\sum_{v \in V} b^\lambda(v) = 0\) are given that define the supply and demand realizations, denoted by \(b = (b^1, \ldots, b^{\lambda_0})\). A vertex with a positive or negative balance, that is, with a supply or demand, is termed source or sink, respectively. In general, the source (sink) vertices do not necessarily have to be the same in all scenarios. If none of the sources (sinks) is reachable from any other source (sink), we say that the problem has parallel sources (sinks). If each scenario has only one vertex with a positive (negative) balance, we refer to this source (sink) as single source (sink). If the single sources (sinks) of all scenarios are defined by the same vertex, we say that the problem has a unique source (sink). Overall, we obtain the network \((G = (V, A = A_{\text{fix}} \cup A_{\text{free}}), c, b)\).

For a single scenario \(\lambda \in \Lambda\), a \(b^\lambda\)-flow in digraph \(G\) is defined by a function \(f^\lambda : A \rightarrow \mathbb{Z}_{\geq 0}\) that satisfies the flow balance constraints

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

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

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

For the entire set \(\Lambda\) of scenarios, a robust \(b\)-flow \(f = (f^1, \ldots, f^{\lambda_0})\) is defined by a \(\vert \Lambda \vert\)-tuple of integral \(b^\lambda\)-flows \(f^\lambda : A \rightarrow \mathbb{Z}_{\geq 0}\) 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 \(b\)-flow \(f\) is defined by

\[c(f) = \max_{\lambda \in \Lambda} c(f^\lambda).\]

Finally, the ROBT≡ problem is defined as follows.

**Definition 1** (ROBT≡ problem). Given a network \((G = (V, A = A_{\text{fix}} \cup A_{\text{free}}), c, b)\), the robust transshipment problem under consistent flow constraints aims at a robust \(b\)-flow \(f = (f^1, \ldots, f^{\lambda_0})\) of minimum cost.
Further related work

We note that in the case of a single scenario, that is, \(|\Lambda| = 1\), the ROBT\(\equiv\) problem corresponds to the transshipment problem [15]. Analogous to the ROBMCF\(\equiv\) problem, we stress that the Integral Flow Theorem of Dantzig and Fulkerson [15] does not hold for the ROBT\(\equiv\) problem. Although the balances are integral, in general, the solution of the continuous relaxation of the ROBT\(\equiv\) problem is not integral, as shown in the following example.

**Example.** For a set of two scenarios \(\Lambda = \{1, 2\}\), let a ROBT\(\equiv\) instance \((G, c, b)\) be given. Digraph \(G\), the arc cost \(c\), and the nonzero balances \(b\) are visualized in Figure 1. An optimal integral robust \(b\)-flow \(f = (f^1, f^2)\) is given for instance as follows. The first scenario flow \(f^1\) sends one unit along arc \((s, t^1)\) and the second scenario flow \(f^2\) sends one unit along arc \((s, t^2)\). The cost is \(c(f) = 4\) as \(c(f^1) = 0\) and \(c(f^2) = 4\) hold. However, if the integral flow requirement is neglected, an optimal robust \(b\)-flow \(f = (f^1, f^2)\) is given as follows. The first scenario flow \(f^1\) sends half a unit each along arc \((s, t^1)\) and path \(s\) to \(t^1\). The second scenario flow \(f^2\) sends half a unit along arc \((s, t^2)\) and path \(s\) to \(t^2\). The cost is \(c(f) = 3\) as \(c(f^1) = 0.5 \cdot 0 + 0.5 \cdot (2 + 2) = 2\) and \(c(f^2) = 0.5 \cdot 4 + 0.5 \cdot (2 + 0) = 3\) hold.

If integral flows were not required, the ROBT\(\equiv\) problem could be solved in polynomial time by a linear program.

Before concluding this section, we present some related results of the capacitated version of the ROBT\(\equiv\) problem, the ROBMCF\(\equiv\) problem, analyzed in our previous work [9]. We prove the \(\mathcal{NP}\)-hardness of the ROBMCF\(\equiv\) problem on acyclic, SP, and pearl digraphs. For the special case of a constant number of scenarios, we propose a pseudo-polynomial algorithm based on dynamic programming on SP digraphs. For the special case of a unique source and unique sink, we present a polynomial-time algorithm on SP digraphs. We note that the complexity results of the ROBMCF\(\equiv\) problem are not transferable to the ROBT\(\equiv\) problem as they rely on the arc capacities of the network. However, on the basis of our previous work [9], we can provide the following three structural results for the ROBT\(\equiv\) problem. First, if the values to be sent along fixed arcs are given, the ROBT\(\equiv\) problem can be solved in polynomial time by \(|\Lambda|\) separate transshipment problems. Second, if the number of fixed arcs is constant, the ROBT\(\equiv\) problem can be solved in polynomial time by a mixed integer program with a corresponding constant number of integer variables by the algorithm of Lenstra [18]. Third, if the ROBT\(\equiv\) problem has a unique source and a unique sink, the cost of an optimal robust flow is determined by the maximum of the cost of the scenario with the maximum supply and the cost of the scenario with minimum supply. For more details and proofs, we refer to our previous work [9]. Moreover, three further results are provided in Sections 2.3, 3.1.2, and 3.2.

### 2.2 Further related work

In the literature, a variety of logistic applications are represented by minimum cost flow (MCF) models. Adapted to specific applications, several extensions of the MCF or transshipment problem are analyzed. For instance, Seedig [24] introduces the minimum cost flow with minimum quantities problem (MCFMQ). In addition to the requirements of the MCF problem, the flow on all outgoing arcs of the source must either take the value zero or a given minimum quantity. Krumke and Thießen [17] generalize the MCFMQ problem such that the minimum quantity property holds for all arcs. Another extension of the MCF problem is the MCF version of the maximum flow problem with disjunctive constraints studied by Pferschy and Schauer [22].

A maximum flow is sought whose flow is positive for at least one arc of every arc pair included in a predetermined arc set. Due to the equal flow requirement of the ROBT\(\equiv\) problem, we focus on literature with equal flow requirements in the following.

There are several extensions to the maximum flow (MF) and MCF problem which consider equal flow requirements on specified arc sets. For instance, the integral flow with homologous arcs problem (HOMIF), introduced by Sahni [23], aims at a maximum flow whose flow is equal on specified arcs. Sahni proves the \(\mathcal{NP}\)-hardness of the problem. The MCF version of the HOMIF problem is known as the (integer) equal flow problem (EF). By standard techniques, the complexity results of the HOMIF problem can be transferred to the EF problem [1]. Meyers and Schulz [19] discuss the uncapacitated version of the EF problem. They prove the strong \(\mathcal{NP}\)-hardness of the problem even in the case of a single source and sink. In addition, they show that there exists no \(2^{\epsilon(n-1)}\)-approximation algorithm for any fixed \(\epsilon > 0\) (on a digraph with \(n\) vertices), even if a nontrivial solution is guaranteed to exist, unless \(P = \mathcal{NP}\). There are further studies for both the HOMIF and EF problem in the literature [11, 20, 25]. Ali et al. [2] investigate a special case of the EF problem where the specified arc sets have cardinality two. An integral MCF is sought whose flow is equal on a predetermined set of arc pairs. The problem finds application in, for example, crew
scheduling [12]. Therefore, Ali et al. present a heuristic algorithm based on Lagrangian relaxation. Meyers and Schulz [19] refer to this special case as paired integer equal flow problem (PEFP). They also consider the uncapacitated version of the PEFP problem and prove the strong $\mathcal{NP}$-hardness. Furthermore, they prove that there exists no $2^{\left(1-\varepsilon\right)}$-approximation algorithm for any fixed $\varepsilon > 0$ on a digraph with $n$ vertices, unless $P = \mathcal{NP}$. The statement holds true even if a nontrivial solution is guaranteed to exist. Hassin and Poznanski [13] analyze the integrality of the multinetwn min-cost equal-flow problem (MEFP) problem and prove the $\mathcal{NP}$-hardness for acyclic digraphs.

Unlike the research referenced above, we do not consider demand and supply for only one scenario in the ROBT $\equiv$ problem. Like in the ROBMC$F$ problem, we consider several scenarios to represent demand uncertainty. We stress that the equal flow requirements are only of importance for more than one scenario. The flow on a fixed arc has to be equal among all scenarios. In turn, the flow on different fixed arcs need not be the same in different scenarios. Despite several scenarios, the problem of finding a feasible solution to the ROBT $\equiv$ problem can be modeled as a special case of the EF, PEF, and MEF problem by means of digraph copies. The problem of finding a feasible solution to the ROBT $\equiv$ problem cannot be modeled as the SEF problem, except for the special case of one fixed arc only. However, these correspondences only apply to finding a feasible solution due to different objectives. We note that the complexity results are not transferable as the ROBT $\equiv$ problem is a special case of the EFP, SEFP, and PEFP problem.

To the best of our knowledge, equal flow requirements and demand uncertainty are combined in our previous study [9] for the first time. Demand uncertainty is frequently studied in the context of (uncapacitated) robust network design. The aim is to decide on capacities such that the entire demand can simultaneously be routed in all considered scenarios while the cost for installing the capacities is supposed to be minimized. Cacchiani et al. [10] present a branch-and-cut algorithm for two types of uncertainty sets, a discrete set of scenarios and a polytope. Altin et al. [3, 4] propose the so-called Hose uncertainty model. Koster et al. [16] consider the budget uncertainty set introduced by Bertsimas and Sim [6, 7].

2.3 First complexity results for acyclic digraphs

In this section, we analyze the complexity of the ROBT $\equiv$ problem on acyclic digraphs. In our previous study [9], we prove the strong $\mathcal{NP}$-completeness of the ROBMC$F$ problem on acyclic digraphs by a reduction from the $(3, B2)$-SAT problem. The $(3, B2)$-SAT problem is a special case of the 3-SAT problem [14] where every literal occurs exactly twice [5]. However, the result is not transferable as the arc capacities are indispensable for the construction of the reduction. Adjusting this construction, we perform a similar reduction as shown in the proof of the following theorem.

**Theorem 1.** Deciding whether or not a feasible solution exists to the ROBT $\equiv$ problem on acyclic digraphs is strongly $\mathcal{NP}$-complete, even if only two scenarios are considered that have the same unique source and unique sink.

A proof can be found in Appendix A. Before concluding this section, we point out the difference between the reductions of the ROBMC$F$ and ROBT $\equiv$ problem. In the ROBMC$F$ problem, we control that one flow unit is sent via every clause vertex by means of arc capacities. In the ROBT $\equiv$ problem, we guarantee this by means of successive fixed arcs included in the additional integrated path $\tilde{p}$ in combination with a scenario in which only one unit is sent.

3 Complexity for SP digraphs

In this section, we analyze the complexity of the ROBT $\equiv$ problem on SP digraphs. In Section 3.1, we prove that the problem is in general weakly $\mathcal{NP}$-complete. In Section 3.2, we provide a polynomial-time algorithm for the special case of networks with a unique source and a unique sink. In Section 3.3, we provide a polynomial-time algorithm for the special case of networks with a unique source and parallel sinks or parallel sources and a unique sink. In the following sections, we use the notation $[n] := \{1, \ldots, n\}$.

Based on the edge SP multi-graphs definition of Valdes et al. [26], we define SP digraphs as follows.

**Definition 2** (SP digraph). An SP digraph is 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 origin of digraph $G$ is $o$, and the target is $q$.

The series and parallel compositions are illustrated in Figure 2. In general, SP digraphs are multi-digraphs with one definite origin and one definite target. In the following, we denote the origin and target of an SP digraph $G$ by $o_G$ and $q_G$, respectively. An SP digraph can be represented in the form of a rooted binary decomposition tree, a so-called SP tree, as shown in Figure 2b. The SP tree indicates the composition of the SP digraph by three different types of vertices, namely $L$-vertices, $S$-vertices, and $P$-vertices. Each arc of the SP digraph is represented by an individual leaf of the SP tree, an $L$-vertex. The $S$- and $P$-vertices are the SP tree’s inner vertices whose associated subgraphs are obtained by a series or parallel composition, respectively, of the subgraphs associated with their two child vertices. The SP tree can be constructed in polynomial time.

3.1 | Multiple sources and multiple sinks networks

Before discussing the general case of multiple sources and multiple sinks, we analyze the complexity of the $\text{ROBT} \equiv$ problem for the special case of networks based on SP digraphs with a unique source and single sinks. In Section 3.1.1, we present a specific instance of the $\text{ROBT} \equiv$ problem. In Section 3.1.2, we perform, based on this specific instance, a reduction from a weakly $NP$-complete special case of the $\text{PARTITION}$ problem, the $\text{EVEN-ODD PARTITION}$ problem [14]. For the $\text{EVEN-ODD PARTITION}$ problem, positive integer pairs are to be separated by a partition of equal weight. The problem is formally defined as follows.

**Definition 3** (Even-odd partition problem). Let $s_1, \ldots, s_{2n}$ be $2n$ positive integers partitioned in sets $S_i = \{s_{2i-1}, s_{2i}\}$ for $i \in [n]$, referred to as pairs, where the integers sum up to $2w$, that is, $\sum_{j=1}^{2n} s_j = 2w$. The $\text{EVEN-ODD PARTITION}$ problem asks whether there exists a disjoint partition $S^1$, $S^2$ of the integers $s_1, \ldots, s_{2n}$ such that

(i) the sum of all integers is equal in both subsets, that is,

$$\sum_{s_i \in S^1} s_i = \sum_{s_i \in S^2} s_i = w,$$

(ii) each pair is separated, that is,

$$|S^1 \cap S_i| = |S^2 \cap S_i| = 1 \text{ for all } i \in [n].$$

We note that $\sum_{i=1}^{n} s_{2i-1} < 2w$ holds true as we consider positive integers.

3.1.1 | Maximum split instance

In this section, we consider a specific instance of the $\text{ROBT} \equiv$ problem, which we call the maximum split instance. First, we construct the maximum split instance. Second, we present properties of an optimal robust flow of this instance. The main property is of the same name as the instance, namely, an optimal flow maximally splits the units to be sent, that is, no two units are sent along the same path.
Construction of the maximum split instance

For \( n \in \mathbb{Z} \) and \( n \geq 2 \), let \( I \) be an EVEN-ODD PARTITION instance with positive integers \( s_1, \ldots, s_{2n} \) partitioned in pairs \( S_i = \{s_{2i-1}, s_{2i}\}, \; i \in [n] \) such that \( \sum_{i=1}^{2n} s_i = 2w \) holds. Without loss of generality, we assume the pairs to be given such that \( s_{2i-1} \geq s_{2i} \) holds for \( i \in [n] \). Considering a set of two scenarios \( \Lambda = \{1, 2\} \), we construct the corresponding maximum split instance \( I_n = (G_n, c, b) \) as visualized in Figure 3.

Let \( G_n = (V_n, A_n) \) be an SP digraph with vertex set \( V_n \) and arc set \( A_n \). The vertex set \( V_n \) contains auxiliary vertices \( v_1, \ldots, v_{2n-1} \) and three specified vertices \( s, t_1, t_2 \). The arc set \( A_n \) contains four types of arcs, termed as detour, cross, shortcut, and blocked arcs. The detour arcs are \( a^d_1 = (s, v_1), a^d_{i+1} = (v_i, v_{i+1}) \) for \( i \in [2n-2] \), and \( a^d_{2n} = (v_{2n-1}, t_1) \). They create a path from vertex \( s \) via auxiliary vertices \( v_1, \ldots, v_{2n-1} \) to vertex \( t_1 \). The cross arcs are \( a^c_1 = (s, v_1) \) and \( a^c_{i+1} = (v_i, v_{i+1}) \) for \( i \in [2n-2] \). They create a path from vertex \( s \) via auxiliary vertices \( v_1, \ldots, v_{2n-2} \) to vertex \( v_{2n-1} \). The shortcut arcs are \( a^s_1 = (s, t_2) \) and \( a^s_{i+1} = (v_i, t_2) \) for \( i \in [2n-1] \). They create a direct connection between vertices \( s, v_1, \ldots, v_{2n-1} \) and vertex \( t_2 \). The single blocked arc \( a^b = (t_1, t_2) \) connects vertices \( t_1 \) and \( t_2 \). We identify the cross arcs as fixed arcs contained in set \( A^f_n \) and all other arcs as free arcs contained in set \( A^a_n \) such that \( A_n = A^f_n \cup A^a_n \) holds. We note that the digraph includes the parallel arcs \( a^c_i \) and \( a^d_i \) for \( i \in [2n-1] \) which are defined by the same vertex pair for simplicity’s sake. In the following, we do not address the arcs by their vertex pairs but by their individual designation. Furthermore, we note that digraph \( G_n \) is an SP digraph as it can be constructed recursively by series and parallel compositions only. Initially, detour arc \( a^d_1 \) is serially composed with the single blocked arc \( a^b \) at vertex \( t_1 \) and this composition in turn is composed in parallel with shortcut arc \( a^s_2 \). Subsequently, the digraph is built up for all \( i \in [2n-1] \) as follows. First, the two parallel arcs \( a^d_{2n-1}, a^d_{2n-2} \) are serially composed with the already constructed digraph at vertex \( v_{2n-1} \). Second, the resulting composition is composed in parallel with arc \( a^s_{2n-1} \).

For all arcs \( a \in A_n \), we define the cost \( c \) as follows. The cost of detour arcs \( a^d_i \) for \( i \in [2n] \) is

\[
    c(a^d_i) = \begin{cases} 
        s_i - s_{i+1} & \text{if } i \in \{1, 3, \ldots, 2n-1\}, \\
        s_i - s_{i+1} + 2^{n-i}nw & \text{if } i \in \{2, 4, \ldots, 2n-2\}, \\
        s_{2n} + 2^0nw & \text{if } i = 2n.
    \end{cases}
\]

The cost of cross arcs \( a^c_i \) for \( i \in [2n-1] \) is

\[
    c(a^c_i) = \begin{cases} 
        0 & \text{if } i \in \{1, 3, \ldots, 2n-1\}, \\
        2^{n-i-1}nw & \text{if } i \in \{2, 4, \ldots, 2n-2\}.
    \end{cases}
\]

The cost of shortcut arcs \( a^s_i \) for \( i \in [2n] \) is

\[
    c(a^s_i) = \begin{cases} 
        s_{i+1} + Fw & \text{if } i \in \{1, 3, \ldots, 2n-1\}, \\
        s_{i-1} + Fw & \text{if } i \in \{2, 4, \ldots, 2n\},
    \end{cases}
\]

with \( F = 2^{n+1} - n - 2 \).
The cost of the blocked arc is \( c(d^i) = 2^{n+1} n^3 w \). We note that the cost is non-negative for all arcs as \( s_{2i-1} \geq s_{2i} \) holds for \( i \in [n] \) and \( s_{2i-1} \leq 2w \leq 2^{n-i} nw \) holds for \( i \in [n-1] \). For all vertices \( v \in V_n \), we define the balances \( b = (b^1, b^2) \) by

\[
b^1(v) = \begin{cases} 
  n & \text{if } v = s, \\
  -n & \text{if } v = t_1, \\
  0 & \text{otherwise},
\end{cases} \quad b^2(v) = \begin{cases} 
  n & \text{if } v = t_2, \\
  -n & \text{if } v = t_1, \\
  0 & \text{otherwise}.
\end{cases}
\]

Vertex \( s \) specifies the unique source and vertices \( t_1 \) and \( t_2 \) specify the sinks of the first and second scenario, respectively. Overall, we obtain a feasible \( \text{ROBT} \equiv \) instance \( I_n = (G_n, c, b) \).

Properties of an optimal robust flow of the maximum split instance

In this paragraph, we present some properties of an optimal robust \( b \)-flow \( f = (f^1, f^2) \) of the maximum split instance \( I_n = (G_n, c, b), n \geq 2 \). As a result, we establish a unique structure of an optimal robust flow which is independent of the \( \text{EVEN-ODD PARTITION} \) instance on which the maximum split instance is based. The first property is that the flow on the fixed arcs decreases with decreasing distance to vertex \( v_{2n-1} \), as shown in the following Lemma. A proof is found in Appendix B.

**Lemma 2.** For an optimal robust \( b \)-flow \( f = (f^1, f^2) \) of the maximum split instance \( I_n = (G_n, c, b) \), it holds

\[
f^1(a^i_1) = f^2(a^i_1) \geq f^1(a^i_{i+1}) = f^2(a^i_{i+1}) \text{ for all } i \in [2n-2].
\]

The next property states along which paths the scenario flows \( f^1, f^2 \) of an optimal robust \( b \)-flow \( f \) might send units and what cost is incurred. In the first scenario, flow \( f^1 \) may only use cross and detour arcs to reach the sink. Once flow \( f^1 \) sends a unit along a detour arc \( a^d_i \) for \( i \in [2n-1] \), the unit is forwarded along the successive detour arcs \( a^d_{i+1}, \ldots, a^d_{2n} \). If the unit was forwarded along one of the successive cross arcs \( a^c_{i+1}, \ldots, a^c_{2n-1} \). Lemma 2 would not be satisfied. Consequently, in the first scenario units are sent along paths of the form \( p^1_i = sa^d_1a^c_2a^d_3a^c_4 \ldots a^c_{i-1}va^iv_d^i a^d_{i+1} \ldots a^d_{2n-1}va^iv_d^i \) for \( i \in [2n] \) as visualized in Figure 4. The costs of paths \( p^1_{2i-1} \) and \( p^1_{2i} \) for \( i \in [n] \) are

\[
c(p^1_{2i-1}) = \sum_{j=1}^{2i-2} c(a^f_j) + \sum_{j=2i-1}^{2n} c(a^f_j)
= \sum_{j=1}^{i-1} 2^{n-j}nw + \sum_{j=i}^{n} (s_{2j-1} - s_{2j}) + \sum_{j=i}^{n-1} (s_{2j} - s_{2j+1} + 2^{n-j}nw) + s_{2n} + 2^0 nw
= nw(2^i - 2)2^{n-i-1} + s_{2i-1} + nw(2^{n-i} - 1)
= nw(2^{n-i} + 2^{n-i} - 1) + s_{2i-1}
\]

and

\[
c(p^1_{2i}) = \sum_{j=1}^{2i-1} c(a^f_j) + \sum_{j=2i}^{2n} c(a^f_j)
\]

\[
= \sum_{j=1}^{i-1} 2^{n-j}nw + \sum_{j=i+1}^{n} (s_{2j-1} - s_{2j}) + \sum_{j=i}^{n-1} (s_{2j} - s_{2j+1} + 2^{n-j}nw) + s_{2n} + 2^0 nw
= nw(2^i - 2)2^{n-i-1} + s_{2i} + nw(2^{n-i+1} - 1)
= nw(2^{n-i} + 2^{n-i-1} - 1) + s_{2i},
\]

respectively. Overall, the cost of the first scenario’s paths \( p^1_i, i \in [2n] \) is \( c(p^1_i) = nw(2^{n-\lceil \frac{i}{2} \rceil} + 2^{n-1} - 1) + s_i \).

In the second scenario, flow \( f^2 \) only uses cross and shortcut arcs to reach the sink. The blocked arc \( d^i \) is not used due to its high cost compared to the (sum of) other arcs. The detour arcs \( a^d_i, i \in [2n-1] \) are not used, because they cost about twice as much as the cross arcs (plus \( s_i \) minus \( s_{i+1} \)). Moreover, the use of cross instead of detour arcs reduces not only the cost in the second but also in the first scenario. The detour arc \( a^d_{2n} \) is not used, otherwise the blocked arc \( d^i \) would subsequently have to be used to reach the sink. Consequently, in the second scenario units are sent along paths of the form \( p^2_i = sa^c_1a^c_2a^c_3a^c_4 \ldots a^c_{i-1}va^iv_d^i a^c_{i+1} \) for \( i \in [2n] \) as visualized in Figure 4. The costs of paths \( p^2_{2i-1} \) and \( p^2_{2i} \) for \( i \in [n] \) are

\[
c(p^2_{2i-1}) = \sum_{j=1}^{2i-2} c(a^f_j) + c(a^d_{2i-1}) = \sum_{j=1}^{i-1} 2^{n-j}nw + s_{2i} + Fw = nw(2^i - 2)2^{n-i-1} + s_{2i} + Fw
\]
FIGURE 4  Example of paths $p^1_i$ and $p^2_i$ for $i \in \{1, 3, \ldots, 2n - 1\}$ and their dependence on each other.

and

$$c(p^2_i) = \sum_{j=1}^{2i-1} c(a^f_j) + c(a^s_j) = \sum_{j=1}^{i-1} 2^{n-j-1} nw + s_{2i-1} + Fw = nw(2^i - 2)2^{n-i} + s_{2i-1} + Fw,$$

respectively. Overall, the cost of the second scenario’s paths $p^2_i$, $i \in \{2n\}$ is

$$c(p^2_i) = \begin{cases} nw(2^i - 2)2^{n-i} + s_{2i-1} + Fw & \text{with } i \in \{1, 3, \ldots, 2n - 1\}, \\
2w(2^i - 2)2^{n-i} + s_{2i-1} + Fw & \text{with } i \in \{2, 4, \ldots, 2n\}. \end{cases}$$

We note that paths $p^1_i$ and $p^2_i$ include the same fixed arcs for all $i \in \{2n\}$. Thus, the choice of the paths in the first and second scenario depends on each other due to the consistent flow constraints. More precisely, if a unit is sent along path $p^1_i$, $i \in \{2n\}$ in the first scenario, a unit is also sent along path $p^2_i$ in the second scenario (and vice versa), as visualized in Figure 4. If the unit was sent along a different path $p \neq p^2_i$ in the second scenario, the flow would be either infeasible as the consistent flow constraints are not satisfied (for $p$ with $\{a^1_1, \ldots, a^1_{i-1}\} \notin A(p)$ or $a^1_i \notin A(p)$ for at least one $j \in \{i, \ldots, 2n - 1\}$) and/or the unit could be sent cheaper (for $p$ with $a^1_i \in A(p)$ for at least one $j \in \{2n - 1\}$). We refer to this property as equal paths property. In the next step, we note the following about the cost of the scenario flows. In terms of minimizing the cost in the first scenario, the aim is a flow which uses as few detour arcs as possible or, in other words, as many cross arcs as possible. In terms of minimizing the cost in the second scenario, the aim is a flow which uses as few cross arcs as possible. The best case for the first scenario’s cost and the worst case for the second scenario’s cost occur if a robust flow sends all units along path $p^1_{2n}$ in the first and along path $p^2_{2n}$ in the second scenario. Conversely, the worst case for the first scenario’s cost and the best case for the second scenario’s cost occur if a robust flow sends all units along path $p^1_1$ in the first and along path $p^2_1$ in the second scenario. As the cost of a robust flow is the maximum of its scenario flows’ cost, we aim at a trade-off between the best and worst case of the first and second scenario’s cost. The fixed (cross) arcs are used in both scenarios and contribute significantly to the cost incurred. Therefore, in the following lemma, we consider the cost of an optimal robust flow incurred on the fixed arcs. A proof is found in Appendix B.

**Lemma 3.** For an optimal robust $b$-flow $f = (f^1, f^2)$ of the maximum split instance $I_n = (G_n = (V_n, A_n), c, b)$ with $A_n = A_n^{\text{free}} \cup A_n^{\text{fix}}$, the total cost incurred on the fixed arcs is $c^{\text{fix}} := nw(2^{n-1} n - 2^n + 1)$, that is,

$$\sum_{a \in A_n^{\text{fix}}} c(a) \cdot f^1(a) = \sum_{a \in A_n^{\text{fix}}} c(a) \cdot f^2(a) = c^{\text{fix}}.$$

As a result, the flow on the fixed arcs is determined for an optimal robust flow as shown in the following lemma. A proof is found in Appendix B.

**Lemma 4.** For an optimal robust $b$-flow $f = (f^1, f^2)$ of the maximum split instance $I_n = (G_n = (V_n, A_n), c, b)$ with $A_n = A_n^{\text{free}} \cup A_n^{\text{fix}}$, the flow on the fixed arcs is given by $f^1(a^s_{i+1} \in \{n-i, n-i+1\}) = f^2(a^s_{i+1}) \in \{n-i, n-i+1\}$ for $a^s_{i+1} \in A_n^{\text{fix}}$ with $i \in [n]$ and $f^1(a^s_i) = f^2(a^s_i) = n - i$ for $a^s_i \in A_n^{\text{fix}}$ with $i \in [n-1]$.

Using the properties about the paths and their cost as well as the insights of Lemmas 2–4, we present the structure of an optimal robust $b$-flow $f = (f^1, f^2)$ of the maximum split instance $I_n$, visualized in Figure 5 (where the shortcut arcs are only hinted).

As $f^1(a^s_i) = n - i$ holds for $i \in [n - 1]$, we obtain $f^1(a^s_i) = i$ for $i \in [n - 1]$. Due to the flow balance constraints, it holds $f^1(a^s_{i-1}) \in \{i - 1, i\}$ for $i \in [n]$. As $f^2(a^s_i) = n - i$ holds for $i \in [n - 1]$, we obtain $f^2(a^s_i) = 1$ for either shortcut arc $a = a^s_{i-1}$ or $a = a^s_2$ with $i \in [n]$. We note that each shortcut arc is included in only one path of the second scenario. Thus, flow $f^2$ sends one unit along either path $p^2_{2n-1}$ or $p^2_{2n}$ for all $i \in [n]$. Because of the equal path property, flow $f^1$ also sends one unit along either path $p^1_{2n-1}$ or $p^1_{2n}$ for all $i \in [n]$. We refer to this property as one path of a pair property. Overall, we obtain the unique structure of an optimal robust flow where one unit is sent along one path of each pair in both scenarios.
Finally, we conclude with remarks about the design of the arc cost of the maximum split instance. By choosing powers of two as arc cost, we achieve that a differentially sent unit—compared to the unique structure—immediately incurs cost (in the first or second scenario) so high that it cannot be compensated for afterwards. Furthermore, we choose the arc cost (excluding integers $s_i$, $i \in [2n]$) sufficiently large, in particular greater than the sum of all integers, that is, $2w = \sum_{i=1}^{2n} s_i \leq nw \leq Fw$. As a result, the unique structure of an optimal robust flow is independent of the EVEN-ODD PARTITION instance. We note that the unique structure of an optimal robust flow meets the desired trade-off between the best and worst case of the first and second scenario’s cost. If the integers included in the arc cost are neglected, the significantly higher remaining cost is equal in both scenarios of an optimal robust flow as shown in the following. By Lemma 3, the cost incurred on the fixed arcs is $c^{\text{fix}}$ in both scenarios. If the integers included in the cost of the free (detour and shortcut) arcs are neglected, the remaining cost incurred on the free arcs is $c^{\text{free int}} := nw(2^{n+1} - n - 2)$ in both scenarios. The $n$-times use of shortcut arcs in the second scenario (as $f^2(a) = 1$ for either $a = a_{2i-1}$ or $a = a_{2i}$ with $i \in [n]$) causes the same cost as the use of detour arc $a_i^1$ once, plus detour arc $a_i^2$ twice, and so on, plus the $n$-times use of detour arc $a_i^2$ in the first scenario (as $f^1(a_{2i}^2) = 0$ for $i \in [n]$), that is,

$$n \cdot Fw = c^{\text{free int}} = 1 \cdot 2^{n-1}nw + 2 \cdot 2^{n-2}nw + \ldots + n \cdot 2^0 nw.$$  

This relation explains why we set parameter $F$ to the value $2^{n+1} - n - 2$. We note that the multiplication of the powers of two by $n$ is necessary for the determination of parameter $F$ to ensure the integrality of the arc cost. Overall, we obtain a lower bound on the cost of an optimal robust flow by $c(f) \geq c(f^1) \geq c^{\text{fix}} + c^{\text{free int}}$, $\lambda \in \Lambda$.

### 3.1.2 Special case of unique source and single sinks networks

In this section, we analyze the complexity of the ROBT $\equiv$ problem for the special case of networks based on SP digraphs with a unique source and single sinks. Before performing a reduction from the weakly $NP$-complete EVEN-ODD PARTITION problem, we define the decision version of the ROBT $\equiv$ problem as follows.

**Definition 4** (Decision version RobT $\equiv$ problem). The decision version of the ROBT $\equiv$ problem asks whether a robust flow exists with cost at most $\beta \in \mathbb{Z}_{\geq 0}$.

**Theorem 5.** The decision version of the RobT $\equiv$ problem is weakly $NP$-complete for networks based on SP digraphs with a unique source and single sinks, even if only two scenarios are considered.

**Proof.** Let $I$ be an EVEN-ODD PARTITION instance with positive integers $s_1$, …, $s_{2n}$ partitioned in pairs $S_i = \{s_{2i-1}, s_{2i}\}$ with $s_{2i-1} \geq s_{2i}$ for $i \in [n]$ such that $\sum_{j=1}^{2n} s_j = 2w$ holds. We consider the corresponding maximum split instance $I_n = (G_n = (V_n, A_n, c, b))$ as visualized in Figure 3. In the following, we prove that $I$ is a Yes-instance if and only if there exists a robust $b$-flow for ROBT $\equiv$ instance $I_n$ with cost of at most $\beta := w + c^{\text{fix}} + c^{\text{free int}} = w + (2^{n-1}n - 2^n + 1 + F)nw$.

Let $S^1, S^2$ be a feasible partition for instance $I$. We determine a robust $b$-flow $f = (f^1, f^2)$ as follows. For every integer pair $S_i = \{s_{2i-1}, s_{2i}\}, i \in [n]$ separated such that $s_{2i} \in S^1$ and $s_{2i-1} \in S^2$ hold, we define for all arcs $a \in A_n$ the flows

$f^1_{\ell_{2i}}(a) = \begin{cases} 1 & \text{for } a \in A(p_{2i}^1), \\ 0 & \text{otherwise,} \end{cases}$

$f^2_{\ell_{2i}}(a) = \begin{cases} 1 & \text{for } a \in A(p_{2i}^2), \\ 0 & \text{otherwise.} \end{cases}$

For every integer pair $S_i = \{s_{2i-1}, s_{2i}\}, i \in [n]$ separated such that $s_{2i-1} \in S^1$ and $s_{2i} \in S^2$ hold, we define for all arcs $a \in A_n$ the flows

$f^1_{\ell_{2i-1}}(a) = \begin{cases} 1 & \text{for } a \in A(p_{2i-1}^1), \\ 0 & \text{otherwise,} \end{cases}$

$f^2_{\ell_{2i-1}}(a) = \begin{cases} 1 & \text{for } a \in A(p_{2i-1}^2), \\ 0 & \text{otherwise.} \end{cases}$
Using these flows, we construct the first and second scenario flow by \( f^1 = \sum_{s_{i} \in S^1} f_{s_{i}}^1 \) and \( f^2 = \sum_{s_{i} \in S^2} f_{s_{i}}^2 = \sum_{s_{i,j} \in S^2} f_{s_{i,j}}^2 + \sum_{s_{i} \in S^2} f_{s_{i,i}}^2 \), respectively. As each integer pair \( S_i = \{s_{2i-1}, s_{2i} \}, i \in [n] \) is separated in partition \( S^1, S^2 \), flow \( f^i, \lambda \in \Lambda \) sends one unit along either path \( p^i_{s_{i},j} \) or \( p^i_{s_{i},i} \) for all \( i \in [n] \). Overall, flows \( f^1 \) and \( f^2 \) send \( n \) units each from the source to their sinks and their values are equal on the fixed arcs. Consequently, robust flow \( f = (f^1, f^2) \) is feasible as the flow balance and consistent flow constraints are satisfied. For computing the cost of flows \( f^1 \) and \( f^2 \), we use that set \( S^1 \) as well as set \( S^2 \) contains one integer of each pair \( S_i = \{s_{2i-1}, s_{2i} \}, i \in [n] \). We obtain

\[
c(f^1) = \sum_{s_{i} \in S^1} c(f_{s_{i}}^1) = \sum_{s_{i} \in S^1} \left( nw \left( 2^{n-1} i + 2^{n-1} - 1 \right) + s_i \right) = \sum_{s_{i} \in S^1} s_i + nw \sum_{i=1}^{n} \left( 2^{n-1} i + 2^{n-1} - 1 \right)
\]

\[
= w + (2^{n-1} - n + 2^{n-1} - 1)nw = w + (2^{n-1} - n + 2^{n-1} - 2 - 2 + 1)nw = w + (2^{n-1} - n + 2^{n-1} + F)nw
\]

\[
= w + c_{\text{fix}} + c_{\text{free}} + c_{\text{int}}
\]

and

\[
c(f^2) = \sum_{s_{2i-1} \in S^2} c(f_{s_{2i-1}}^2) + \sum_{s_{2i} \in S^2} c(f_{s_{2i}}^2)
\]

\[
= \sum_{s_{2i-1} \in S^2} \left( nw(2^i - 2)2^{n-1-i} + s_{2i-1} + Fw \right) + \sum_{s_{2i} \in S^2} \left( nw(2^i - 2)2^{n-1-i} + s_{2i} + Fw \right)
\]

\[
= \sum_{s_{i} \in S^2} s_i + nw \sum_{i=1}^{n} \left( (2^i - 2)2^{n-1-i} + Fw \right)
\]

\[
= w + (2^{n-1} - n + 2^{n-1} + F)nw
\]

\[
= w + c_{\text{fix}} + c_{\text{free}} + c_{\text{int}}.
\]

We have constructed a robust \( b \)-flow \( f = (f^1, f^2) \) for ROBT instance \( I_n \) with cost \( c(f) = w + c_{\text{fix}} + c_{\text{free}} = \beta \).

Conversely, let \( f = (f^1, f^2) \) be a robust \( b \)-flow for ROBT instance \( I_n \) with cost \( c(f) = \max \{ c(f^1), c(f^2) \} \leq \beta \). We may assume that flow \( f \) satisfies the unique structure of an optimal robust \( b \)-flow of the maximum split instance \( I_n \), where a lower bound on the cost is given by \( c(f) \geq c_{\text{fix}} + c_{\text{free}} \). If flow \( f \) sent one unit differently, it would use at least one other detrour or shortcut arc or would not use at least one of the detrour or shortcut arcs. Due to the structure of the SP digraph, flow \( f \) would be different on the fixed arcs and the cost \( c(f) \) would exceed the value \( \beta = w + c_{\text{fix}} + c_{\text{free}} \). We note that the gap between lower bound \( c_{\text{fix}} + c_{\text{free}} \) and upper bound \( \beta \) on the cost of flow \( f \) is \( w \). In the following, we determine the cost of the scenario flows \( f^1 \) and \( f^2 \). Let \( f^i_{\lambda} \) denote the value which flow \( f^i_{\lambda}, \lambda \in \Lambda \) sends along path \( p^i_{\lambda}, i \in [2n] \). Using Lemma 4, we obtain for the cost of the first scenario

\[
c(f^1) = \sum_{a \in \mathcal{A}} c(a) \cdot f^1(a)
\]

\[
= \sum_{i=1}^{2n-1} c(a_{i}^1) \cdot f^1(a_{i}^1) + \sum_{i=1}^{2n} c(a_{i}^0) \cdot f^1(a_{i}^0)
\]

\[
= \sum_{i=1}^{n-1} 2^{n-1-i}nw \cdot f^1(a_{2i}^1) + \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^1) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1} + 2^{n-1} - nw) \cdot f^1(a_{2i}^1) + (s_{2n} + 2^{n} - nw) \cdot f^1(a_{2n}^1)
\]

\[
= \sum_{i=1}^{n-1} 2^{n-1-i}nw \cdot (n - i) + \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^1) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1} + 2^{n-1} - nw) \cdot i + (s_{2n} + 2^{n} - nw) \cdot n
\]

\[
= \sum_{i=1}^{n-1} 2^{n-1-i}nw \cdot (n - i) + \sum_{i=1}^{n} 2^{n-1-i}nw \cdot i + \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^1) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1}) \cdot i + s_{2n} \cdot n
\]

\[
= c_{\text{fix}} + c_{\text{free}} + \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^1) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1}) \cdot i + s_{2n} \cdot n.
\]

Considering the last part of the expression above, we define \( a := \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^1) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1}) \cdot i + s_{2n} \cdot n \). Before simplifying term \( a \), we note the following. Due to the one path of a pair property, it holds \( f^1(p^i_{s_{i}}} = 1 - f^1(p^i_{s_{i}}) = 0 \) for all \( i \in [n] \). Furthermore, it holds \( f^1(a_{2i-1}^1) \in \{i - 2, i\} \), where \( f^1(a_{2i-1}^1) = i - 1 \)
if \( f^1(p_{2i-1}^1) = 0 \) and \( f^1(a_{2i-1}^1) = i \) if \( f^1(p_{2i-1}^1) = 1 \). Overall, we obtain \( f^1(a_{2i-1}^2) = i - 1 + f^1(p_{2i-1}^1) \) for \( i \in [n] \). Using this as well as Lemma 4 and the one path of a pair property, we perform the following transformation

\[
\alpha = \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot f^1(a_{2i-1}^2) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1}) \cdot i + s_{2n} \cdot n \\
= \sum_{i=1}^{n} (s_{2i-1} - s_{2i}) \cdot (i - 1 + f^1(p_{2i-1}^1)) + \sum_{i=1}^{n-1} (s_{2i} - s_{2i+1}) \cdot i + s_{2n} \cdot n \\
= \sum_{i=1}^{n} (s_{2i-1} \cdot f^1(p_{2i-1}^1) + s_{2i} \cdot (1 - f^1(p_{2i-1}^1))) \\
= \sum_{i=1}^{n} (s_{2i-1} \cdot f^1(p_{2i-1}^1) + s_{2i} \cdot f^1(p_{2i}^1)) \\
= \sum_{i=1}^{2n} s_i \cdot f^1(p_i^1).
\]

Overall, we obtain

\[
c(f^1) = c^\text{fix} + c^\text{free} + \sum_{i=1}^{2n} s_i \cdot f^1(p_i^1).
\]

Using Lemma 4, the one path of a pair property, and the fact that \( f^2(a_i^1) = f^2(p_i^1) \) holds for all \( i \in [2n] \), we obtain for the cost of the second scenario

\[
c(f^2) = \sum_{a_i \in A_a} c(a) \cdot f^2(a) \\
= \sum_{i=1}^{2n-1} c(a_i^1) \cdot f^2(a_i^1) + \sum_{i=1}^{n} c(a_i^2) \cdot f^2(a_i^2) \\
= \sum_{i=1}^{n-1} 2^{n-1} n \cdot f^2(a_i^2) + \sum_{i=1}^{n} ((s_{2i-1} + Fw) \cdot f^2(a_i^2) + (s_{2i} + Fw) \cdot f^2(a_i^2)) \\
= \sum_{i=1}^{n-1} 2^{n-1} n \cdot (n - i) + n \cdot Fw + \sum_{i=1}^{n} (s_{2i-1} \cdot f^2(p_{2i}^2) + s_{2i} \cdot f^2(p_{2i}^2)) \\
= c^\text{fix} + c^\text{free} + \sum_{i=1}^{n} (s_{2i-1} \cdot f^2(p_{2i}^2) + s_{2i} \cdot f^2(p_{2i}^2)).
\]

As \( c(f^2) \leq \beta = c^\text{fix} + c^\text{free} + w \) holds for \( \lambda \in \Lambda \), we obtain

\[
\sum_{i=1}^{2n} s_i \cdot f^1(p_i^1) \leq w \tag{1}
\]

and

\[
\sum_{i=1}^{n} (s_{2i-1} \cdot f^2(p_{2i}^2) + s_{2i} \cdot f^2(p_{2i}^2)) \leq w. \tag{2}
\]

In the next step, we define a partition \( S^1, S^2 \) by

\[
S^1 = \{s_{2i-1} | f^1(p_{2i-1}^1) = 1, \ i \in [n]\} \cup \{s_{2i} | f^1(p_{2i}^1) = 1, \ i \in [n]\},
\]

\[
S^2 = \{s_{2i} | f^2(p_{2i}^2) = 1, \ i \in [n]\} \cup \{s_{2i-1} | f^2(p_{2i-1}^2) = 1, \ i \in [n]\}.
\]

Due to the equal paths and the one path of a pair property, either \( f^1(p_{2i-1}^1) = f^2(p_{2i-1}^2) = 1 \) or \( f^1(p_{2i}^1) = f^2(p_{2i}^2) = 1 \) holds for all \( i \in [n] \). For this reason, neither subset \( S^1 \) nor \( S^2 \) includes both integers of an integer pair. Instead, by definition of the sets, one integer of every integer pair \( S_i = \{s_{2i-1}, s_{2i}\}, i \in [n] \) is included in set \( S^1 \) and the other in set \( S^2 \). Consequently, partition \( S^1, S^2 \) contains the positive integers \( s_1, \ldots, s_{2n} \); it is disjoint, and it separates all integer pairs. Expressions (1) and (2) are equivalent to expressions

\[
\sum_{s_i \in S^1} s_i \leq w \quad \text{and} \quad \sum_{s_i \in S^2} s_i \leq w.
\]
respectively. Thus, it holds
\[ \sum_{s_i \in S^1} s_i = \sum_{s_i \in S^2} s_i = w. \]

Finally, we have constructed a feasible partition \( S^1, S^2 \) for the EVEN-ODD PARTITION instance \( I \).

As a result, we obtain the following corollary.

**Corollary 6.** The decision version of the RobT\(\equiv\) problem is weakly \( \mathcal{NP} \)-complete for networks based on SP digraphs with multiple sources and multiple sinks, even if only two scenarios are considered.

In the special case of a constant number of scenarios, we can solve the ROBTo problem for networks based on SP digraphs with multiple sources and multiple sinks by the pseudo-polynomial-time algorithm presented in our previous work [9]. The algorithm is based on dynamic programming. The runtime of the algorithm depends on the arc capacities required as input. For the ROBTo problem, we may set the capacity of every arc to the maximum total supply among all scenarios.

### 3.2 Special case of unique source and unique sink networks

In this section, we analyze the complexity of the ROBTo problem for the special case of networks based on SP digraphs with a unique source and a unique sink. Let \( (G, c, b) \) be such a ROBTo instance on SP digraph \( G = (V, A = A^{\text{fix}} \cup A^{\text{free}}) \) with unique source \( s \) and unique sink \( t \). We note that SP digraphs are acyclic which is why vertices can be omitted that are not reachable from the unique source or from which the unique sink is not reachable. Thus, without loss of generality, we assume that the unique source \( s \) and unique sink \( t \) coincide with the origin and target of SP digraph \( G \), respectively. Further, we assume that the scenarios are non-decreasingly ordered with respect to the supply of the unique source, that is, \( b^i(s) \leq \ldots \leq b^{i|\Lambda|}(s) \).

In our previous study [9], we present a polynomial-time algorithm (Algorithm 5.1) to solve the ROBMCF\(\equiv\) problem for the special case of networks based on SP digraphs with a unique source and a unique sink. If we choose sufficiently large arc capacities, we can use the algorithm to solve the ROBTo problem. We obtain the following corollary.

**Corollary 7.** The ROBTo problem is solvable in polynomial time on SP digraphs with a unique source and a unique sink.

We note the following about the Algorithm 5.1 of our previous work [9]. The algorithm uses that there exists an optimal robust \( b \)-flow that sends in each scenario \( \lambda \in \Lambda \) the minimum of the number of units which the arc capacities allow and the so-called excess supply \( b^\lambda(s) - \min_{\lambda' \in \Lambda} b^{\lambda'}(s) = b^\lambda(s) - b^1(s) \) along a shortest path (with respect to arc cost \( c \) in digraph \( G - A^{\text{fix}} \)). In our case, the total excess supply of each scenario can be sent along a single shortest path in digraph \( G - A^{\text{fix}} \). As the remaining supply that needs to be sent in each scenario is equal \( (b^1(s) \text{ units}) \), it may be sent through the entire SP digraph \( G \) at minimum cost. Consequently, the algorithm reduces to the computation of two shortest paths—one in SP digraph \( G \) and one in digraph \( G - A^{\text{fix}} \)—along which flow is sent. For an alternative algorithm, we refer to our preliminary version of this paper [8]. The alternative algorithm provides further insights into the ROBTo problem for networks based on SP digraphs with a unique source and a unique sink. It uses the SP structure to shrink the SP digraph to only one multi-arc on which the ROBTo problem is solved. Before concluding this section, we present a new result about the structure of an existing optimal robust flow, which we use in the next section.

**Lemma 8.** Let \( I = (G, c, b) \) be a ROBTo instance on SP digraph \( G = (V, A = A^{\text{fix}} \cup A^{\text{free}}) \) with unique source \( s \) and unique sink \( t \). There exists an optimal robust \( b \)-flow \( \tilde{f} = (\tilde{f}^1, \ldots, \tilde{f}^{|\Lambda|}) \) such that for all feasible robust \( b \)-flows \( f = (f^1, \ldots, f^{|\Lambda|}) \) it holds \( c(\tilde{f}^\lambda) \leq c(f^\lambda) \) for all \( \lambda \in \Lambda \).

**Proof.** Let \( p^{\text{fix}} \) and \( p^{\text{free}} \) be shortest \((s, t)\)-paths in digraphs \( G \) and \( G - A^{\text{fix}} \), respectively. If \( c(p^{\text{free}}) \leq c(p^{\text{fix}}) \) holds, the feasible robust \( b \)-flow that sends in each scenario the entire supply along path \( p^{\text{free}} \) is optimal in the uncapacitated unique source, unique sink network and it fulfills the desired property. Thus, in the following, we consider the case \( c(p^{\text{free}}) > c(p^{\text{fix}}) \). Using Algorithm 5.1 of our previous work [9], we obtain an optimal robust \( b \)-flow \( \tilde{f} = (\tilde{f}^1, \ldots, \tilde{f}^{|\Lambda|}) \) defined as follows

\[
\tilde{f}^\lambda(a) = \begin{cases} 
  b^\lambda(s) & \text{for arcs } a \in A(p^{\text{fix}}) \setminus A(p^{\text{free}}), \\
  b^\lambda(s) - b^1(s) & \text{for arcs } a \in A(p^{\text{free}}) \setminus A(p^{\text{fix}}), \\
  b^\lambda(s) & \text{for arcs } a \in A(p^{\text{fix}}) \cap A(p^{\text{free}}), \\
  0 & \text{otherwise}.
\end{cases}
\]
We claim that for all feasible robust $b$-flows $f = (f_1, \ldots, f^{|\Lambda|})$ it holds $c(\hat{f}^{x'}) \leq c(f^x)$ for all $\lambda \in \Lambda$. Assume this does not hold true, that is, there exists a feasible robust $b$-flow $\hat{f} = (\hat{f}_1, \ldots, \hat{f}^{|\Lambda|}) \neq f$ such that $c(\hat{f}^{x'}) < c(f^x)$ holds for at least one scenario $\lambda' \in \Lambda$. Let $\mathcal{P} = (p_1, \ldots, p_k)$ be an $(s, t)$-path decomposition of flow $\hat{f}$, that is, it holds $\hat{f}^{x'} = \sum_{i=1}^k \hat{f}_{[v]}^{x'}$, where $\hat{f}_{[v]}^{x'}$ indicates the part of flow $\hat{f}$ restricted to path $p_i$. We assume that the paths are ordered such that the first $r$ paths may contain fixed and free arcs and the last $k-r$ paths contain free arcs only, that is, $p_1, \ldots, p_r \subseteq G$ and $p_{r+1}, \ldots, p_k \subseteq G - A^{\mathfrak{f}}$. We note that $c(p_{\mathfrak{f}}) \leq c(p_i)$ holds for all $i \in [r]$ and $c(p_{\mathfrak{f}}) \leq c(p_i)$ holds for all $j \in \{r+1, \ldots, k\}$. Every scenario flow $\hat{f}^{x'}, \lambda \in \Lambda$ of a feasible robust $b$-flow $f$ sends at least $b(s) - b(t)$ units in digraph $G - A^{\mathfrak{f}}$ due to the consistent flow constraints. This means that the value $\sum_{i=r+1}^k \hat{f}_{[v]}^{x'}(p_j)$ of the flow $\sum_{j=r+1}^k \hat{f}_{[v]}^{x'}$ is at least $b(s) - b(t)$. If the value of the flow $\sum_{j=r+1}^k \hat{f}_{[v]}^{x'}(p_j)$ is greater than or equal to $b(s) - b(t)$, the value of flow $\sum_{i=1}^r \hat{f}_{[v]}^{x'}(p_i)$ is less than or equal to $b(s)$, respectively. Using the $(s, t)$-path decomposition, we obtain a lower bound on the cost of flow $\hat{f}$ as follows

$$c(\hat{f}^{x'}) = \sum_{p \in b} c(p) \cdot \hat{f}^{x'}(p) = \sum_{i=1}^r c(p_i) \cdot \hat{f}^{x'}(p_i) + \sum_{j=r+1}^k c(p_j) \cdot \hat{f}^{x'}(p_j) \overset{(1)}{\geq} c(p_{\mathfrak{f}}) \cdot \sum_{i=1}^r \hat{f}^{x'}(p_i) + c(p_{\mathfrak{f}}) \cdot \sum_{j=r+1}^k \hat{f}^{x'}(p_j) \overset{(2)}{=} c(p_{\mathfrak{f}}) \cdot b(s) + c(p_{\mathfrak{f}}) \cdot (b(s) - b(t)) = c(f^{x'})$$

As we consider a robust $b$-flow $\hat{f} \neq f$, at least one of the inequalities (1), (2) is strict. We obtain a contradiction to the assumption.

### 3.3 | Special cases of unique source and parallel sinks and parallel sources and unique sink networks

In this section, we analyze the complexity of the RobT+ problem for the special case of networks based on SP digraphs with a unique source and parallel (multiple) sinks. First, we introduce terms and notations. Subsequently, we present structural results for a feasible robust flow arising from the considered special case. Finally, based on these structural results, we provide a polynomial-time algorithm. By replacing all arcs by their reverse arcs and identifying the source as sink and the sinks as sources, the results can be analogously applied for the special case of networks based on SP digraphs with parallel (multiple) sources and a unique sink.

Let $(G, c, b)$ be a RobT+ instance on SP digraph $G$ with a unique source and parallel sinks. Analogous to the previous section we assume without loss of generality that the unique source, denoted by $s$, coincides with the origin $o_G$ of the SP digraph $G$. Let $\gamma(\lambda)$ indicate the total number of sinks in scenario $\lambda \in \Lambda$. The $j$-th sink of scenario $\lambda \in \Lambda$ is specified by $t^j_\lambda$. We say a subgraph $G_{vw} \subseteq G$ is spanned by two vertices $v, w \in V(G)$ (for which a $(v, w)$-path exists) if subgraph $G_{vw}$ is induced by vertices reachable from vertex $v$ and from which vertex $w$ is reachable, that is, $V(G_{vw}) = \{ x \in V(G) \mid \text{there exist a } (v, x)-\text{and a } (x, w)-\text{path} \}$. A spanned subgraph $G_{vw}$ of an SP digraph is an SP digraph itself with origin $v$ and target $w$ as shown in Lemma D1 in Appendix D.

In the next step, we define vertex labels $\pi : V(G) \rightarrow \mathbb{2}^{V(G)}$ that indicate at every vertex $v \in V(G)$ the reachable sinks, that is, $\pi(v) = \{ t \in V(G) \mid t \text{ is a sink and there exists a}(v, t)\text{-path})$. As all vertices are reachable from the origin in an SP digraph, the label of the unique source is given by the set of all sinks, that is, $\pi(s) = \{ t^j_\lambda \mid \lambda \in \Lambda \}$. We note that the labels form inclusion chains from the unique source $s$ to the sinks $t^j_\lambda$, $\lambda \in \Lambda$. We say two labels $\pi(v), \pi(w)$ of vertices $v, w \in V(G)$ are $(v, w)$-consecutive if $\pi(w) \subseteq \pi(v)$ holds and there does not exist a vertex $z \in V(G)$ with label $\pi(z)$ such that $\pi(w) \not\subseteq \pi(z) \subseteq \pi(v)$ holds. Furthermore, we say a vertex $v \in V(G)$ is a last vertex (of label $\pi(v)$) if

(i) its label $\pi(v)$ is nonempty,

(ii) for each vertex $w \in V(G)$ for which an arc $a = (v, w) \in A(G)$ exists the labels $\pi(v), \pi(w)$ are $(v, w)$-consecutive.

As the labels are inclusion chains, a last vertex $v \in V(G)$ with label $\pi(v)$ implies that for each vertex $w \in V(G)$ for which a $(v, w)$-path exists it holds $\pi(w) \not\subseteq \pi(v)$. A last vertex of a label is uniquely determined as shown in the following lemma.
by the following vertex and arc set

\[ \pi(o_{G_x}) = (t^1, t^2, t^3) \]

(B) Subgraph \( G_x \) for \( x \in V(T) \) of SP tree \( T \) is the smallest subgraph that contains sinks \( \pi(v) = \pi(w) \) and its origin \( o_{G_x} \) is a last vertex with label \( \pi(o_{G_x}) = (t^1, t^2, t^3) \).

Lemma 9. Let \( (G, c, b) \) be a ROB\(T\) instance on SP digraph \( G \) with a unique source and parallel sinks. For each label, there is a uniquely determined last vertex.

Proof. Firstly, we note that the label of each sink contains only the sink itself. Clearly, for the label of a sink, the uniquely determined last vertex is the sink itself. As the labels of the sinks are the only one element labels, the statement is true for all labels with cardinality one. Assume now the statement is false for a label of cardinality at least two. There exist at least two last vertices \( v, w \in V(G) \) of the same label \( \pi(v) = \pi(w) \) with \( |\pi(v)| = |\pi(w)| \geq 2 \). The vertices are parallel, otherwise we immediately obtain a contradiction to the definition of a last vertex. If we consider an SP tree \( T \) of SP digraph \( G \), we see that there exist \( P \)-vertices by which the sinks \( t^i, i \in [\gamma_T(\lambda)], \lambda \in \Lambda \) are composed in parallel. We note that if two subgraphs, each containing one sink, are composed in parallel, the two sinks may not be the origin or target of the composition. Otherwise, there exists a path between two sinks which contradicts the definition of parallel sinks. Let \( G_x \) denote the subgraph of SP digraph \( G \) that is associated to vertex \( v \in V(T) \) of SP tree \( T \), that is, subgraph \( G_x \) contains all arcs which are represented by the leaves of SP tree \( T \) below vertex \( v \). There exists a \( P \)-vertex \( x \in V(T) \) whose associated subgraph \( G_x \subseteq G \) is in a bottom-up search in SP tree \( T \) the first subgraph that contains the parallel sinks of set \( \pi(v) = \pi(w) \). More precisely, for \( x \in V(T) \) it holds \( \pi(v) = \pi(w) \subseteq V(G_x) \) but for all descendants \( y \in V(T) \) of vertex \( x \) it holds \( \pi(v) = \pi(w) \not\subseteq V(G_y) \). Furthermore, we note that none of the sinks complies with the origin and target of subgraph \( G_x \), that is, \( t^i \neq o_{G_x} \) and \( t^i \neq q_{G_x}, i \in [\gamma_T(\lambda)], \lambda \in \Lambda \). An example is visualized in Figure 6. If we further follow the SP tree’s instructions to compose SP digraph \( G \), we continue composing subgraphs serially at the origin or target or in parallel at the origin and target of subgraph \( G_x \). Consequently, the access from the unique source \( s \) to the sinks of set \( \pi(v) = \pi(w) \) is determined only by the origin \( o_{G_x} \) of subgraph \( G_x \). By choice of subgraph \( G_x \), origin \( o_{G_x} \) is a last vertex of the same label as the last vertices \( v, w \), that is, \( \pi(o_{G_x}) = \pi(v) = \pi(w) \). Thus, vertices \( v, w \) need to be parallel to origin \( o_{G_x} \), as visualized in Figure 6b, or one of them coincides with origin \( o_{G_x} \). In both cases, we obtain a contradiction as the access of the sinks is only possible via origin \( o_{G_x} \) as discussed above.

We save the label structure by means of a directed tree referred to as label tree. The label tree \( T^\pi \) of SP digraph \( G \) is defined by the the following vertex and arc set

\[ V(T^\pi) = \{ s \} \cup \{ v \in V(G) \mid v \text{ is a last vertex} \}, \]

\[ A(T^\pi) = \{(s, w) \mid w \in V(T^\pi) \text{ and } \pi(s) = \pi(w) \} \cup \{(v, w) \mid v, w \in V(T^\pi) \text{ and vertices } v, w \text{ are consecutive} \}. \]

The label tree is a tree spanned by the unique source and all vertices. The root is defined by the unique source \( s \) and the leaves are defined by the sinks \( t^i, i \in [\gamma_T(\lambda)], \lambda \in \Lambda \). An example is visualized in Figure 7. We note that the label tree can be constructed by breadth-first search [15] at every vertex in polynomial time.

Using the label tree, in the following lemma, we present two properties satisfied by every feasible robust flow.

Lemma 10. Let \( I = (G, c, b) \) be a ROB\(T\) instance on SP digraph \( G \) with unique source \( s \) and parallel sinks \( t^i, i \in [\gamma_T(\lambda)], \lambda \in \Lambda \). Further, let \( T^\pi \) be the label tree of SP digraph \( G \), \( G_{vw} \subseteq G \) be the subgraph spanned by the vertices incident to arc \( (v, w) \in A(T^\pi) \), and \( \widetilde{G} := \bigcup_{(v, w) \in A(T^\pi)} G_{vw} \) be the union. For a feasible robust \( b \)-flow \( f = (f^1, \ldots, f^{|A|}) \) the following holds true.
and the composed union of the subgraphs \( \beta \lambda \)

\[ \text{FIGURE 7} \]

For simplicity’s sake, we denote the sum of the demand of sinks in scenario (ii) the value that flows within subgraph \( G_{vw} \)

\[ \sum_{(v,w) \in A(T^*)} \sum_{f(a) > 0} \sum_{t \in T} |b^\lambda(t)|, \ \lambda \in \Lambda. \]

For simplicity’s sake, we denote the sum of the demand of sinks in scenario \( \lambda \in \Lambda \) reachable from vertex \( v \in V(G) \) by parameter \( \beta^\lambda(v) \), that is,

\[ \beta^\lambda(v) = \sum_{t \in T^\lambda} |b^\lambda(t)|. \]

**Proof.** For the proof of the first statement, we note that \( \tilde{G} \subseteq G \) holds true. Furthermore, we note that \( \pi(z) \neq \emptyset \) holds for all vertices \( z \in V(\tilde{G}) \) and \( \pi(z) = \emptyset \) holds for all vertices \( z \in V(G) \setminus V(\tilde{G}) \). For an example, we refer to Figure 7. Assume now the first statement is false. There exists a feasible robust \( b \)-flow \( f = (f^1, \ldots, f^\Lambda) \) with f(a) > 0 for at least one arc \( a = (x, y) \in A(G) \setminus A(\tilde{G}) \) in one scenario \( \lambda \in \Lambda \). For the labels of the incident vertices of arc \( a = (x, y) \) it holds either \( \pi(x) = \emptyset \) or \( \pi(y) = \emptyset \) and \( \pi(y) = \emptyset \) due to the inclusion chains formed by the labels. Thus, in both cases none of the sinks \( t^i, i \in [\gamma(\lambda)], \ \lambda \in \Lambda \) is reachable via arc \( a \). We obtain \( f(a) = 0 \) for all scenarios \( \lambda \in \Lambda \) because of the flow balance constraints, which contradicts the assumption.

For the proof of the second statement, let \( \lambda \in \Lambda \) be an arbitrary scenario. Equality (E1) holds true as the flow which enters SP digraph \( G_{vw} \) via vertex \( v \) needs to reach vertex \( w \). There does not exist a sink in subgraph \( G_{vw} \) (unless vertex \( w \) is a sink itself) due to the following reason. As \( w \in V(T^*) \) holds, we obtain \( \pi(w) \neq \emptyset \) by definition. Thus, either vertex \( w \) is a sink itself or a number of sinks is reachable from vertex \( w \). In both cases, none of the vertices \( z \in V(G_{vw}) \setminus \{w\} \) is in \( \Lambda \) as it would not be parallel to the sinks of set \( \pi(w) \). Equality (E2) holds true as the value of the ingoing flow at vertex \( w \) in SP digraph \( G_{vw} \) must be equal to the sum of the demand of the sinks reachable from vertex \( w \), that is, \( \beta^\lambda(w) \). If the value was greater, the flow balance constraints would not be satisfied due to oversupply. If the value was lower, the demand of the sinks of set \( \pi(w) \) could not be met as the only access is via vertex \( w \). We note that no additional units are sent beyond subgraph \( G_{vw} \) via vertex \( w \) to sinks of set \( \pi(w) \) as there does not exist an arc \( (u, w) \in A(G) \setminus A(G_{vw}) \) due to the following reason. We note that there does not exist a \( (v, u) \)-path with \( u \in V(G) \setminus V(G_{vw}) \). If there existed a \( (v, u) \)-path, it would be included with arc \( (a, w) \) in subgraph \( G_{vw} \). As \( G \) is an SP digraph, there does not exist a \( (u, v) \)-path either due to \( (v, w) \)-path and the assumption of the existence of arc \( (u, w) \). Thus, vertices \( u \) and \( v \) are parallel as visualized in Figure 8. Let \( T \) be an SP tree of SP digraph \( G \). As \( v \) is a last vertex, there must exist a P-vertex \( x \in V(T) \) whose associated subgraph \( G_t \) is the first subgraph that contains all sinks of set \( \pi(v) \) and has vertex \( v \) as origin. Let \( G_1 \) and \( G_2 \) denote the subgraphs by which the subgraph \( G_t \) is composed in parallel. As \( w \) is a last vertex with \( \pi(w) \subseteq \pi(v) \), it is contained in either subgraph \( G_1 \) or \( G_2 \) but it is not the target. If vertex \( w \) was the target, the sinks in set \( \pi(v) \setminus \pi(w) \subseteq V(G_t) \) would not be parallel to the sinks in

\[ \pi(z) = \pi(v) = \{t^1_1, t^2_1, t^3_1, t^4_1, t^5_1\}, \ \pi(v_2) = \pi(v) = \{t^1_2, t^2_2, t^3_2, t^4_2, t^5_2\}, \ \pi(v_3) = \pi(v) = \{t^1_3, t^2_3, t^3_3, t^4_3, t^5_3\}, \ \pi(v_4) = \pi(v) = \{t^1_4, t^2_4\}, \ \pi(v) = \pi(t^2_4) = \{t^2_4\}. \]

\[ \pi(v_2) = \pi(v) = \{t^1_2, t^3_2\}, \ \pi(v_3) = \{t^1_3, t^2_3\}, \ \lambda \in [2], i \in [3], \ \pi(v_i) = \emptyset, i \in (8,9,10,11) \]

\[ \text{FIGURE 7} \quad \text{Example of an SP digraph} \ G, \ \text{vertex labels} \ \pi(v), \ v \in V(G), \ \text{label tree} \ T^*, \ \text{subgraphs spanned by vertices incident to arcs} \ (v_1, v_2), (v_2, t^2_4) \in A(T^*), \ \text{and the composed union of the subgraphs} \ \tilde{G} = \cup_{v \in V(G)} G_{vw}. \]
If vertex $v \in V(G) \setminus V(G_{vw})$ incident to arc $(u, w)$ existed, it would be parallel to vertex $v$.

As a result, we can focus on the spanned subgraphs induced by the label tree’s arcs to compute a robust flow. Furthermore, using Lemma 10, we provide the analogous result to Lemma 8 about the structure of an existing optimal robust flow.

**Lemma 11.** Let $I = (G, c, b)$ be a RobT≡ instance on SP digraph $G$ with a unique source and parallel sinks. There exists an optimal robust $b$-flow $\hat{f} = (\hat{f}_1, \ldots, \hat{f}_{|\Lambda|})$ such that for all feasible robust $b$-flows $f = (f_1, \ldots, f_{|\Lambda|})$ it holds $c(\hat{f}^\lambda) \leq c(f^\lambda)$ for all $\lambda \in \Lambda$.

**Proof.** Let $T^c$ be the label tree of SP digraph $G$ and let $G_{vw}$ be the subgraph spanned by the vertices incident to arc $(v, w) \in A(T^c)$. By Lemma 10, the flow is zero beyond digraph $\tilde{G} = \bigcup_{(v, w) \in A(T^c)} G_{vw}$ and the values which flow within subgraphs $G_{vw}$, $(v, w) \in A(T^c)$ are determined for every feasible robust flow. Consequently, only the flow within a subgraph $G_{vw}$, $(v, w) \in A(T^c)$ may differ in feasible and in particular in optimal solutions. Considering the subgraphs $G_{vw}$ for all $(v, w) \in A(T^c)$ separately, we split RobT≡ instance $I$ into instances $I_{vw} = (G_{vw}, c, b_{vw})$ where the arc cost $c$ on the subgraphs remains equal and the new balances $b_{vw} : V(G_{vw}) \rightarrow \mathbb{Z}$ are defined as follows:

$$b_{vw}^\mu(u) = \begin{cases} 0 & \text{for all } u \in V(G_{vw}) \setminus \{v, w\}, \\ \beta^\lambda(w) & \text{for } u = v, \\ -\beta^\lambda(w) & \text{for } u = w. \end{cases}$$

Instance $I_{vw}$, $(v, w) \in A(T^c)$ is a RobT≡ instance based on an SP digraph with a unique source and a unique sink. By Lemma 8, there exists an optimal robust $b_{vw}$-flow $\tilde{f}_{vw} = (\tilde{f}_{vw}^1, \ldots, \tilde{f}_{vw}^{|\Lambda|})$ such that for all feasible robust $b_{vw}$-flows $f_{vw} = (f_{vw}^1, \ldots, f_{vw}^{|\Lambda|})$ it holds $c(\tilde{f}_{vw}^\lambda) \leq c(f_{vw}^\lambda)$ for all $\lambda \in \Lambda$. The composed flow $\tilde{f} = \sum_{(v, w) \in A(T^c)} \tilde{f}_{vw}$ is a feasible robust $b = \sum_{(v, w) \in A(T^c)} b_{vw}$-flow for RobT≡ instance $I$ as the flow balance and consistent flow constraints are satisfied. Let $\hat{f}$ be an arbitrary feasible robust $b$-flow for RobT≡ instance $I$. Further, let $\tilde{f}_{|G_{vw}}$ be the flow which results from restricting flow $\hat{f}$ to subgraph $G_{vw}$, $(v, w) \in A(T^c)$. Due to Lemma 10, flow $\tilde{f}_{|G_{vw}}$ is a feasible robust $b_{vw}$-flow for instance $I_{vw}$, $(v, w) \in A(T^c)$. Thus, it holds $c(\tilde{f}_{vw}^\lambda) \leq c(\tilde{f}_{vw}^\lambda)$ for all $\lambda \in \Lambda$. We obtain

$$c(\hat{f}^\lambda) = \sum_{(v, w) \in A(T^c)} c(\tilde{f}_{vw}^\lambda) \leq \sum_{(v, w) \in A(T^c)} c(\tilde{f}_{vw}^\lambda) = c(\tilde{f}^\lambda)$$

for all scenarios $\lambda \in \Lambda$, which implies $c(\hat{f}) \leq c(\tilde{f})$. Consequently, robust $b$-flow $\hat{f}$ is optimal and for all feasible robust $b$-flows $f = (f^1, \ldots, f_{|\Lambda|})$ it holds $c(\hat{f}^\lambda) \leq c(f^\lambda)$ for all $\lambda \in \Lambda$. □

Using Lemmas 10 and 11, we specify the composition of an optimal robust flow.

**Lemma 12.** Let $I = (G, c, b)$ be a RobT≡ instance on SP digraph $G$ with a unique source and parallel sinks. Further, let $T^c$ be the label tree of SP digraph $G$ and $G_{vw}$ be the subgraph spanned by the vertices incident to arc $(v, w) \in A(T^c)$. Let $I_{vw} = (G_{vw}, c, b_{vw})$ be the RobT≡ instance restricted to subgraph $G_{vw} \subseteq G$ where the arc cost $c$ remains equal and the new balances are defined as follows:

$$b_{vw}^\mu(u) = \begin{cases} 0 & \text{for all } u \in V(G_{vw}) \setminus \{v, w\}, \\ \beta^\lambda(w) & \text{for } u = v, \\ -\beta^\lambda(w) & \text{for } u = w. \end{cases}$$

For $(v, w) \in A(T^c)$, let $f_{vw} = (f_{vw}^1, \ldots, f_{vw}^{|\Lambda|})$ be an optimal robust $b_{vw}$-flow for instance $I_{vw}$ such that for all feasible robust $b_{vw}$-flows $\tilde{f}_{vw} = (\tilde{f}_{vw}^1, \ldots, \tilde{f}_{vw}^{|\Lambda|})$ it holds $c(\tilde{f}_{vw}^\lambda) \leq c(f_{vw}^\lambda)$ for all $\lambda \in \Lambda$. The composed flow $f = \sum_{(v, w) \in A(T^c)} f_{vw}$ is an optimal robust $b = \sum_{(v, w) \in A(T^c)} b_{vw}$-flow for instance $I$. □
Algorithm 3.1.

Input: $\text{ROBT}$ instance $(G, c, b)$ with a unique source and parallel sinks where $G$ is an SP digraph

Output: Robust minimum cost $b$-flow $f$

Method:

1. Construct label tree $T^*$
2. For all arcs $(v, w) \in A(T^*)$ do
   3. Consider the unique source, unique sink instance $I_{vw} = (G_{vw}, c, b_{vw})$ on subgraph $G_{vw}$ spanned by vertices $v, w$, where the arc cost $c$ remains equal and the new balances are defined for $\lambda \in \Lambda$ as follows
   \[
   b^\lambda_{vw}(u) = \begin{cases} 
   0 & \text{for all } u \in V(G_{vw}) \setminus \{v, w\}, \\
   \beta^\lambda(w) & \text{for } u = v, \\
   -\beta^\lambda(w) & \text{for } u = w
   \end{cases}
   \]
4. Compute an optimal robust $b_{vw}$-flow $f_{vw}$ for instance $I_{vw}$ such that for all feasible robust $b_{vw}$-flows $\tilde{f}$ it holds $c(f^\lambda_{vw}) \leq c(\tilde{f}^\lambda_{vw})$ for all $\lambda \in \Lambda$
5. Set $f := \sum_{(v, w) \in A(T^*)} f_{vw}$
6. Return robust $b$-flow $f = (f^1, \ldots, f^{|\Lambda|})$

Proof. The composed flow $f$ is a feasible robust $b$-flow for instance $I$ as the flow balance and consistent flow constraints are satisfied. Assume $f$ is not optimal for instance $I$. There exists a robust $b$-flow $\tilde{f}$ with less cost, that is, $c(\tilde{f^\lambda}) = \max_{\lambda \in \Lambda} c(\tilde{f}^\lambda) < c(f^\lambda) = \max_{\lambda \in \Lambda} c(f^\lambda) = c(f)$ with $\lambda_1, \lambda_2 \in \Lambda$. Let $\tilde{f}^\lambda_{G_{vw}}$ denote the flow obtained by restricting flow $\tilde{f}$ to subgraph $G_{vw}, (v, w) \in A(T^*)$. Using Lemma 10, we obtain for the cost
\[
\sum_{(v, w) \in A(T^*)} c(f^\lambda_{vw}) = c(\tilde{f}^\lambda_{vw}) < c(f^\lambda_{vw}) = c(\tilde{f}^\lambda_{vw}) = \sum_{(v, w) \in A(T^*)} c(f^\lambda_{vw}),
\]
which is a contradiction as $c(f^\lambda_{vw}) \leq c(\tilde{f}^\lambda_{vw})$ holds for all $(v, w) \in A(T^*)$. If $\lambda_1 \neq \lambda_2$ holds, we obtain
\[
\sum_{(v, w) \in A(T^*)} c(f^\lambda_{vw}) = c(\tilde{f}^\lambda_{vw}) < c(f^\lambda_{vw}) = \sum_{(v, w) \in A(T^*)} c(f^\lambda_{vw}),
\]
which is a contradiction as $c(f^\lambda_{vw}) \leq c(\tilde{f}^\lambda_{vw})$ holds for all $(v, w) \in A(T^*)$.

Based on the presented structural results, we provide Algorithm 3.1 to compute an optimal robust flow.

We obtain the polynomial-time solvability of the ROBT problem for networks based on SP digraphs with a unique source and parallel sinks as shown in the following theorem.

Theorem 13. Let $(G, c, b)$ be a RobT instance on SP digraph $G$ with a unique source and parallel sinks. Algorithm 3.1 computes an optimal robust $b$-flow in $O(|V(G)|^2 + |V(G)| \cdot |A(G)|)$ time.

Proof. The correctness of the algorithm results from Lemma 12. Considering the runtime, we obtain the following. The label tree $T^*$ can be constructed by breadth-first search [15] at every vertex in $O(|V(G)|^2 + |V(G)| \cdot |A(G)|)$ time. For each instance $I_{vw}, (v, w) \in A(T^*)$, an optimal robust $b_{vw}$-flow $f_{vw}$ with the desired property is constructed in $O(|V(G)| + |A(G)|)$ time by means of the methods of Section 3.2. Overall, this can be done in $O(|V(G)|^2 + |V(G)| \cdot |A(G)|)$ time as $A(T^*) \leq V(G)$ holds true. In total, the algorithm runs in $O(|V(G)|^2 + |V(G)| \cdot |A(G)|)$ time.

Finally, we note that the results of this section can be applied to the ROBT problem for networks based on SP digraphs with parallel sources and parallel sinks provided there exists a path between each source and each sink. For this special case, we need to determine two vertices, the first vertex $v_1$ reachable from all sources and the last vertex $v_2$ from which all sinks are reachable.
We note that the first and last vertex exist (they might coincide) as we not only consider parallel sources and parallel sinks but also assume that there exists a path between each source and each sink. Using vertices \( v_1, v_2 \), we divide a ROBT\( \equiv \) instance \( I \) into the three instances \( I_1, I_2, I_3 \)—a parallel sources, unique sink, a unique source, parallel sinks, and a unique source, unique sink instance—as visualized in Figure 9. Instances \( I_1 = (G_{v_1 v_1}, c, b_{v_1 v_1}), I_2 = (G_{v_2 v_2}, c, b_{v_2 v_2}), I_3 = (G_{v_3 v_3}, c, b_{v_3 v_3}) \) are based on the spanned subgraphs \( G_{v_1 v_1}, G_{v_2 v_2}, G_{v_3 v_3} \subseteq G \), respectively. The arc cost \( c \) stays the same and the balances are defined for \( \lambda \in \Lambda \) as follows

\[
\begin{align*}
  b_{v_1 v_1}^k(u) &= \begin{cases} 
    b^k(u) & \text{for all } u \in V(G_{v_1 v_1}) \setminus \{v_1\}, \\
    -\beta^k(v_1) & \text{for } u = v_1,
  \end{cases} \\
  b_{v_2 v_2}^k(u) &= \begin{cases} 
    b^k(u) & \text{for all } u \in V(G_{v_2 v_2}) \setminus \{v_2\}, \\
    \beta^k(v_2) & \text{for } u = v_2,
  \end{cases} \\
  b_{v_3 v_3}^k(u) &= \begin{cases} 
    b^k(u) & \text{for all } u \in V(G_{v_3 v_3}) \setminus \{v_1, v_2\}, \\
    \beta^k(v_1) & \text{for } u = v_1, \\
    -\beta^k(v_2) & \text{for } u = v_2.
  \end{cases}
\end{align*}
\]

For the ROBT\( \equiv \) instances \( I_1, I_2, I_3 \), we determine optimal robust flows \( f_1, f_2, f_3 \), respectively, by means of Algorithm 5.1 of our previous study [9] and Algorithm 3.1. Flows \( f_2 \) and \( f_1, f_2 \) satisfy the properties of Lemmas 8 and 11, respectively. Therefore, analogous to the proof of Lemma 12, we can prove that the composed flow \( f = f_1 + f_2 + f_3 \) is an optimal robust \( b \)-flow for ROBT\( \equiv \) instance \( I \).

4 | COMPLEXITY FOR PEARL DIGRAPHS

In this section, we analyze the complexity of the ROBT\( \equiv \) problem on pearl digraphs—SP digraphs with a specific structure. We note that the ROBMC\( \equiv \) problem is \( \mathcal{NP} \)-complete on pearl digraphs as shown in our previous work [9]. Using the representation of SP trees, we define pearl digraphs based on the definition of Ohst [21].

**Definition 5** (pearl digraph). A pearl digraph is an SP digraph where the corresponding SP tree does not have \( P \)-vertices whose children are \( S \)-vertices.

In other words, a pearl digraph is a path consisting of multi-arcs.

In the following, we use this special structure and shrink every multi-arc of the pearl digraph to a multi-arc consisting of the cheapest fixed and the cheapest free arc (if they exist). As we do not consider arc capacities we may assume that an optimal robust flow only uses the cheapest fixed arc(s) and/or the cheapest free arc(s) of each multi-arc. If this was not the case, the flow could be shifted and sent at lower cost. In addition, we can even remove the remaining fixed arc if a remaining free arc of the same multi-arc exists that is equal or less expensive. For a proof, we refer to the parallel-shrinking procedure of the preliminary version of this paper [8]. Let \( (G, c, b) \) be a ROBT\( \equiv \) instance based on pearl digraph \( G \). We may assume without loss of generality that the vertex and arc set of a pearl digraph \( G = (V, A) \) are of the form \( V = \{v_1, \ldots, v_n\} \) and \( A = \{a_1, \ldots, a_m\} \).
A = \{a_i^\text{fix} = (v_i, v_{i+1}) \mid a_i^\text{free} = (v_i, v_{i+1}) \mid i \in [n-1] \} (in case arcs \(a_i^\text{fix}, a_i^\text{free} \) exist for all \(i \in [n-1] \)). We note that the parallel arcs \(a_i^\text{fix} \) and \(a_i^\text{free} \), \(i \in [n-1] \) form a multi-arc and are defined by the same vertex pair for simplicity’s sake. In the following, we do not address the arcs by their vertex pairs but by their individual designation. The arc set \(A \) is partitioned into sets \(A^\text{fix} \) and \(A^\text{free} \) such that \(a_i^\text{fix} \in A^\text{fix} \) and \(a_i^\text{free} \in A^\text{free} \) hold for all \(i \in [n-1] \). If arcs \(a_i^\text{fix} \) and \(a_i^\text{free} \) exist for \(i \in [n-1] \), we note that \(c(a_i^\text{fix}) < c(a_i^\text{free}) \) holds. An example of a RobT\(\equiv \) instance on a pearl digraph is visualized in Figure 10. Let \(\sigma_i^\text{v} \) denote the sum of the balances of vertices \(v_1, \ldots, v_i \in V \) in scenario \(\lambda \in \Lambda \), i.e., \(\sigma_i^\text{v} = \sum_{j=1}^{i} b^\text{v}(v_j) \). We refer to \(\sigma_i^\text{v} \) as the state of balance, in short state, at vertex \(v_i \in V \) in scenario \(\lambda \in \Lambda \). We note that \(\sigma_i^\text{v} \) units need to be sent from vertex \(v_i \) to vertex \(v_{i+1} \) to satisfy the balance at vertex \(v_i \). Furthermore, we note that state \(\sigma_i^\text{v} \) is non-negative at every vertex \(v_i \in V \) for all scenarios \(\lambda \in \Lambda \). If this was not the case, the total supply of the predecessor vertices \(v_1, \ldots, v_i \) would not meet the total demand of vertices \(v_1, \ldots, v_i \). By parameter \(\sigma_i^\text{min} \), we indicate the minimum state at vertex \(v_i \in V \) among all scenarios \(\lambda \in \Lambda \), i.e., \(\sigma_i^\text{min} = \min_{\lambda \in \Lambda} \sigma_i^\text{v} \). Using the states at the vertices, we formulate the following result about the existence of an optimal robust flow.

**Lemma 13.** Let \(I = (G, c, b) \) be a RobT\(\equiv \)instance on pearl digraph \(G = (V, A = A^\text{fix} \cup A^\text{free}) \). An optimal robust \(b\)-flow is given, if one exists, by \(f = (f^1, \ldots, f^{\mid \Lambda \mid}) \) with scenario flows \(f^\lambda \), \(\lambda \in \Lambda \) defined as follows

\[
f^\lambda(a) = \begin{cases} 
\sigma_{v_i}^\text{v} & \text{for all } a = a_i^\text{fix} \in A^\text{fix} \text{ if } a_i^\text{free} \in A^\text{free} \text{ exists,} \\
\sigma_i^\text{v} - \sigma_i^\text{min} & \text{for all } a = a_i^\text{free} \in A^\text{free} \text{ if } a_i^\text{fix} \in A^\text{fix} \text{ exists,} \\
\sigma_i^\text{v} & \text{for all } a = a_i^\text{fix} \in A^\text{fix} \text{ if there does not exist } a_i^\text{free}, \\
\sigma_i^\text{v} & \text{for all } a = a_i^\text{free} \in A^\text{free} \text{ if there does not exist } a_i^\text{fix}. 
\end{cases}
\]

**Proof.** Flow \(f \) defined as above is a feasible robust \(b\)-flow due to the following two reasons. Firstly, flow \(f^\lambda \), \(\lambda \in \Lambda \) satisfies the flow balance constraints at every vertex \(v_i \in V \) as shown in the following expression

\[
\sum_{a_i \in [a_i^\text{fix}, a_i^\text{free}]} f_i^\lambda(a_i) - \sum_{a_{i-1} \in [a_{i-1}^\text{fix}, a_{i-1}^\text{free}]} f_i^\lambda(a_{i-1}) = \sigma_i^\text{v} - \sigma_{v_{i-1}}^\text{v} = \sum_{j=1}^{i-1} b^\text{v}(v_j) - \sum_{j=1}^{i-1} b^\text{v}(v_j) = b^\text{v}(v_i).
\]

We note that \(\sigma_i^\text{v} = \sigma_i^\text{min} \) must hold true for all scenarios \(\lambda \in \Lambda \) if arc \(a_i^\text{fix} \) exists but arc \(a_i^\text{free} \) does not exist for one \(i \in [n-1] \). Otherwise, there does not exist a feasible robust \(b\)-flow for instance \(I \). Secondly, flows \(f^1, \ldots, f^{\mid \Lambda \mid} \) satisfy the consistent flow constraints as \(\sigma_i^\text{min} \) units are sent along fixed arc \(a_i^\text{fix} \in A^\text{fix} \) in all scenarios \(\lambda \in \Lambda \). Assume flow \(f \) is not optimal. There exists an optimal robust \(b\)-flow \(\tilde{f} = (\tilde{f}^1, \ldots, \tilde{f}^{\mid \Lambda \mid}) \) with less cost, that is, \(c(\tilde{f}) < c(f) \). Due to the consistent flow constraints, it holds \(f^\lambda(a_i^\text{fix}) \leq \sigma_i^\text{min} \) for all fixed arcs \(a_i^\text{fix} \in A^\text{fix} \) and all scenarios \(\lambda \in \Lambda \). There must exist arcs \(a_i^\text{fix}, a_i^\text{free} \in A \) such that \(\tilde{f}^\lambda(a_i^\text{fix}) < \sigma_i^\text{min} \) and \(\tilde{f}^\lambda(a_i^\text{fix}) > \sigma_i^\text{min} \) hold for \(\lambda \in \Lambda \) in order that \(\tilde{f} \) is feasible and \(\tilde{f} \neq f \) holds. We shift as many units of flow \(f^\lambda, \lambda \in \Lambda \) from the free arc \(a_i^\text{free} \) to the fixed arc \(a_i^\text{fix} \) until \(\sigma_i^\text{min} \) units are sent along fixed arc \(a_i^\text{fix} \). As \(c(a_i^\text{free}) > c(a_i^\text{fix}) \) holds, we obtain a feasible robust \(b\)-flow \(\tilde{f} \) with cost \(c(\tilde{f}) < c(f) \), which contradicts the assumption.

Using Lemma 13, we present Algorithm 4.1 to solve the RobT\(\equiv \) problem on pearl digraphs.

We conclude this section with the polynomial-time solvability of the RobT\(\equiv \) problem on pearl digraphs as shown in the following theorem.

**Theorem 14.** Let \((G, c, b) \) be a RobT\(\equiv \)instance on pearl digraph \(G = (V, A = A^\text{fix} \cup A^\text{free}) \). Algorithm 4.1 computes an optimal robust \(b\)-flow in \(O(|V| \cdot |\Lambda|) \) time.
Algorithm 4.1.

Input: ROBT instance \((G,c,b)\) where \(G = (V,A = A^{\text{fix}} \cup A^{\text{free}})\) is a pearl digraph

Output: Robust minimum cost \(b\)-flow \(f\)

Method:

1. Compute state \(\sigma^\lambda_{vi}\) for every vertex \(v_i \in V\) and every scenario \(\lambda \in \Lambda\)
2. Compute the minimum state \(\sigma^\min_{vi}\) for every vertex \(v_i \in V\)
3. For every scenario \(\lambda \in \Lambda\) construct the flow
   \[
   f^\lambda(a) = \begin{cases} 
   \sigma^\min_{vi} & \text{for all } a = a^{\text{fix}}_i \in A^{\text{fix}} \text{ if } a^{\text{free}}_i \in A^{\text{free}} \text{ exists,} \\
   \sigma^\lambda_{vi} - \sigma^\min_{vi} & \text{for all } a = a^{\text{free}}_i \in A^{\text{free}} \text{ if } a^{\text{fix}}_i \in A^{\text{fix}} \text{ exists,} \\
   \sigma^\lambda_{vi} & \text{for all } a = a^{\text{fix}}_i \in A^{\text{fix}} \text{ if there does not exist } a^{\text{free}}_i, \\
   \sigma^\lambda_{vi} & \text{for all } a = a^{\text{free}}_i \in A^{\text{free}} \text{ if there does not exist } a^{\text{fix}}_i 
   \end{cases}
   \]
4. \textbf{return} Robust \(b\)-flow \(f = (f^1, \ldots, f^{|\Lambda|})\)

\textbf{Proof.} The correctness of the algorithm results from Lemma 13. Considering the runtime, we obtain the following. The states can be computed at all vertices in \(O(|V| \cdot |\Lambda|)\) time. The minimum state can be computed for all vertices in \(O(|V| \cdot |\Lambda|)\) time. The construction of the flow is done in \(O(|A| \cdot |\Lambda|)\) time. In total, the algorithm runs in \(O(|V| \cdot |\Lambda|)\) time.

\section{CONCLUSION}

In this article, we considered the ROBT\(\equiv\) problem, the uncapacitated version of the ROBMCF\(\equiv\) problem. As the complexity results of the ROBMCF\(\equiv\) problem depend on the arc capacities given, we analyzed the complexity of the ROBT\(\equiv\) problem. On SP digraphs, we proved the following results. We proved the weak \(\mathcal{NP}\)-completeness, even if only two scenarios are considered that have the same unique source and single (but different) sinks. For the special case of a unique source and parallel sinks or parallel sources and a unique sink, we presented a polynomial-time algorithm. Furthermore, we observed that the algorithm is extendable for the special case of parallel sources and parallel sinks if there exists a path between each source and each sink. On pearl digraphs, we presented a polynomial-time algorithm, independent of the number of sources and sinks. On the basis of our previous work [9], we provided some structural results and provided the following three results. On acyclic digraphs, finding a feasible solution to the ROBT\(\equiv\) problem is strongly \(\mathcal{NP}\)-complete, even if only two scenarios are considered that have the same unique source and unique sink. For the special case of a constant number of scenarios, we observed the pseudo-polynomial-time solvability on SP digraphs. For the special case of a unique source and a unique sink, we presented a polynomial-time algorithm on SP digraphs.

For the future work, we will consider the ROBT\(\equiv\) problem on SP digraphs when the number of scenarios is part of the input. Furthermore, we will study the problem for further graph classes such as digraphs with bounded treewidth.

ACKNOWLEDGMENTS

We would like to recognize the invaluable assistance of Anna Margarethe Limbach who contributed to the proof of Theorem C1 in Appendix C. Open Access funding enabled and organized by Projekt DEAL.

FUNDING INFORMATION

This study was supported by Freigeist-Fellowship of Volkswagen Stiftung, the German research council (DFG) Research Training Group 2236 UnRAVeL, and the German research council (DFG) Collaborative Research Centres SFB 1481 (Sparsity and Singular Structures).

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.
REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, New Jersey, 1993.
[2] A. I. Ali, J. Kennington, and B. Shetty, The equal flow problem, Eur. J. Oper. Res. 36 (1988), 107–115.
[3] A. Altin, E. Amaldi, P. Belotti, and M. Č. Pinar, Provisioning virtual private networks under traffic uncertainty, Networks 49 (2007), 100–155.
[4] A. Altin, H. Yaman, and M. Pinar, The Robust Network Loading Problem Under Hose Demand Uncertainty: Formulation, Polyhedral Analysis, and Computations, INFORMS J. Comput. 23 (2011), 75–89.
[5] P. Berman, M. Karpinski, and A. Scott, Approximation hardness of short symmetric instances of max-3sat, Tech. report, 2004.
[6] D. Bertsimas and M. Sim, Robust discrete optimization and network flows, Math. Program. 98 (2003), 49–71.
[7] D. Bertsimas and M. Sim, The Price of Robustness, Oper. Res. 52 (2004), 35–53.
[8] C. Büsing, A. Koster, and S. Schmitz, On the complexity of robust transshipment under consistent flow constraints, INOC (2022), 1–6.
[9] C. Büsing, A. M. C. A. Koster, and S. Schmitz, Robust minimum cost flow problem under consistent flow constraints, Ann. Oper. Res. 312 (2021), 1–32.
[10] V. Cacchiani, M. Jünger, F. Liers, A. Lodi, and D. R. Schmidt, Single-commodity robust network design with finite and hose demand sets, Math. Program. 157 (2016), 297–342.
[11] H. I. Calvete, Network simplex algorithm for the general equal flow problem, Eur. J. Oper. Res. 150 (2003), 585–600.
[12] P. Carrara E and G. Gallo, Network models for vehicle and crew scheduling, Eur. J. Oper. Res. 16 (1984), 139–151.
[13] R. Hassin and R. Poznanski, Integrality in the min-cost flow problem, Networks 80 (2022), 267–273.
[14] D. S. Johnson and M. R. Garey, Computers and intractability: A guide to the theory of NP-completeness, WH Freeman, New York, 1979.
[15] B. Korte, J. Vygen, B. Korte, and J. Vygen, Combinatorial optimization, Vol 2, Springer, Berlin, 2012.
[16] A. M. C. A. Koster, M. Kutschka, and C. Raack, Robust network design: Formulations, valid inequalities, and computations, Networks 61 (2013), 128–149.
[17] S. O. Krumke and C. Thielen, Minimum cost flows with minimum quantities, Inf. Process. Lett. 111 (2011), 533–537.
[18] H. W. Lenstra Jr., Integer programming with a fixed number of variables, Math. Oper. Res. 8 (1983), 538–548.
[19] C. A. Meyers and A. S. Schulz, Integer equal flows, Oper. Res. Lett. 37 (2009), 245–249.
[20] D. R. Morrison, J. J. Sauppe, and S. H. Jacobson, A network simplex algorithm for the equal flow problem on a generalized network, INFORMS J. Comput. 25 (2013), 2–12.
[21] J. P. Ohst, On the construction of optimal paths from flows and the analysis of evacuation scenarios, Ph.D. thesis, University Koblenz-Landau, Germany, 2015.
[22] U. Pferschy and J. Schauer, The maximum flow problem with disjunctive constraints, J. Comb. Optim. 26 (2013), 109–119.
[23] S. Sahni, Computationally related problems, SIAM J. Comput. 3 (1974), 262–279.
[24] H.G. Seedig, "Operations Research Proceedings 2010," Network flow optimization with minimum quantities, Springer, Berlin, Heidelberg, 2011, pp. 295–300.
[25] K. Srinathan, P. R. Goundan, M. V. N. A. Kumar, R. Nandakumar, and C. P. Rangan, “Computing and combinatorics,” Theory of equal-flows in networks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 514–524.
[26] J. Valdes, R. E. Tarjan, and E. L. Lawler, The recognition of series parallel digraphs, SIAM J. Comput. 11 (1982), 298–313.

How to cite this article: C. Büsing, A. M. C. A. Koster, and S. Schmitz, Robust transshipment problem under consistent flow constraints, Networks. 83 (2024), 53–82. https://doi.org/10.1002/net.22184

APPENDIX A

In the following proof, we use the notation \([n] := \{1, \ldots, n\}\).

**Proof of Theorem 1.** The \(\text{ROBT}^\equiv\) problem is contained in \(\mathcal{NP}\) as we can check in polynomial time whether the flow balance and consistent flow constraints are satisfied for every scenario. Let \(\{x_1, \ldots, x_n\}\) be the set of variables and \(C_1, \ldots, C_m\) be the clauses of the \((3, B2)\)-SAT instance \(I\). For a set of two scenarios \(\Lambda = \{1, 2\}\), we construct a \(\text{ROBT}^\equiv\) instance \(\tilde{I} = (G, c, b)\). Figure A1 visualizes a \(\text{ROBT}^\equiv\) instance corresponding to an example of a \((3, B2)\)-SAT instance with four clauses and three variables. In general, the \(\text{ROBT}^\equiv\) instance is based on a digraph \(G = (V, A)\) defined as follows. The vertex set \(V\) includes one vertex \(v_i\) per variable \(x_i, i \in [n]\), one dummy vertex \(v_{n+1}\), and one vertex \(w_j\) per clause \(C_j, j \in [m]\). For every literal \(x_i\) and \(\overline{x_i}\), \(i \in [n]\), four auxiliary vertices \(w^+_{i\ell} (\overline{w}^-_{i\ell})\), \(\ell \in [4]\) are included. Furthermore, \(V\) includes one auxiliary vertex \(t\) and vertices \(r_{\ell'}\) for \(\ell' \in [2(2n + 2)]\).

In the following proof, we use the notation \([n] := \{1, \ldots, n\}\).
\( p_i \) each with the literals. More precisely, let \( x^k_i (\bar{x}_i^k) \) denote literal \( x_i, (\bar{x}_i) \), \( i \in [n] \) which occurs the \( k \)-th time, \( k \in [2] \) in the formula. Arc \((w_i^{2k-1}, w_i^{2k+1}) (\bar{w}_i^{2k-1}, \bar{w}_i^{2k+1}) \) corresponds to literal \( x^k_i (\bar{x}_i^k), i \in [n], k \in [2] \), referred to as literal arc. Using this correspondence, we add arc \((w_i^{2k}, u_j) ((\bar{w}_i^{2k}, u_j)) \) for every literal \( x^k_i (\bar{x}_i^k), i \in [n], k \in [2] \) included in clause \( C_j, j \in [m] \). In the next step, we create a path \( \tilde{p} \) from vertex \( v_{n+1} \) along vertices \( r_0, \ell' \in [2(2n + 2) - 1] \) to vertex \( z := r_{2(2n+2)} \). Before introducing the last arcs included in arc set \( A \), we identify all literal arcs and each arc in even position of path \( \tilde{p} \) as the only fixed arcs in the network, that is,

\[
A^{\text{fix}} = \{(w_i^e, w_i^{e+1}), (\bar{w}_i^e, \bar{w}_i^{e+1}) | \ell' \in \{1, 3 \}, i \in [n] \} \cup \{(r_\ell, r_{\ell+1}) | \ell' \in \{1, 3, \ldots, 2(2n + 2) - 1\} \}.
\]

We add arcs that connect vertex \( v_1 \) with each tail of all literal arcs and each head of all literal arcs with auxiliary vertex \( t \), that is, \((v_1, w_1^e), (v_1, \bar{w}_1^e) \) for \( \ell' \in \{1, 3\} \) and \((w_i^e, t), (\bar{w}_i^e, t) \) for \( \ell' \in \{2, 4\} \). The clause vertices are connected with the first \( m \) fixed arcs of path \( \tilde{p} \), that is, \((u_j, r_{2j-1}) \) for all \( j \in [m] \). The auxiliary vertex \( t \) is connected with each tail of the successive \( 2n - m \) fixed arcs of path \( \tilde{p} \) by \((t, r_{2j-1}), \ell' \in \{m + 1, \ldots, 2n\} \). We add arcs \((v_1, w_1^e), (v_1, \bar{w}_1^e), (w_i^e, r_{4n+1}), \) and \((\bar{w}_i^e, r_{4n+3}) \). Finally, we connect all \( 2n + 2 \) fixed arcs of path \( \tilde{p} \) with vertex \( z \), that is, \((r_\ell, z) \) for all \( \ell' \in \{2, 4, \ldots, 4n + 2\} \). We set the cost \( c \equiv 0 \) and define the balances \( b = (b^1, b^2) \) by

\[
b^1(v) = \begin{cases} 1 & \text{if } v = v_1, \\ -1 & \text{if } v = z, \\ 0 & \text{otherwise}, \end{cases}
b^2(v) = \begin{cases} 2n + 2 & \text{if } v = v_1, \\ -(2n + 2) & \text{if } v = z, \\ 0 & \text{otherwise}. \end{cases}
\]

Vertices \( v_1 \) and \( z \) specify the unique source and unique sink, respectively. Overall, we obtain a feasible \( \text{ROBT}^\equiv \) instance \( \tilde{I} = (G, c, b) \) which is constructed in polynomial time. Hence, it remains to show that \( I \) is a \( \text{Yes} \)-instance if and only if a feasible robust \( b \)-flow exists for instance \( \tilde{I} \).

Let \( x_1, \ldots, x_n \) be a satisfying truth assignment for instance \( I \). We define the first scenario flow \( f^1 \) of instance \( \tilde{I} \) as follows

\[
f^1(a) = \begin{cases} 1 & \text{for all } a \in A(p_i) \text{ if } x_i = \text{True}, \\ 1 & \text{for all } a \in A(\bar{p}_i) \text{ if } x_i = \text{False}, \\ 1 & \text{for all } a \in A(\bar{p}), \\ 0 & \text{otherwise}. \end{cases}
\]

Flow \( f^1 \) uses either path \( p_i \) or \( \bar{p}_i, i \in [n] \) to send one unit from source \( v_1 \) to vertex \( v_{n+1} \). The unit is forwarded from vertex \( v_{n+1} \) to sink \( z \) along path \( \tilde{p} \). As \( x_1, \ldots, x_n \) is a satisfying truth assignment, there exists one designated verifying literal \( x^k_i \) or \( \bar{x}_i^k, i \in [n], k \in [2] \) for each clause \( C_j, j \in [m] \). Using this, we define the first part of the second scenario flow \( f^2 \) as follows

\[
f^2(a) = 1 \text{ for all } \begin{cases} a \in A(q_i^k) \text{ with } q_i^k = v_1 w_1^{2k-1} w_1^{2k+1} u_j r_{2j-1} r_{2j} z \text{if } x_i \in C_j \text{ is designated as verifying literal}, \\ a \in A(q_i^k) \text{ with } q_i^k = v_1 \bar{w}_1^{2k-1} \bar{w}_1^{2k+1} u_j r_{2j-1} r_{2j} z \text{if } \bar{x}_i \in C_j \text{ is designated as verifying literal}. \end{cases}
\]

Flow \( f^2 \) sends \( m \) units from the source \( v_1 \) to the clause vertices \( u_1, \ldots, u_m \) along the literal arcs corresponding to the designated verifying literals. The \( m \) units are forwarded along the subsequent fixed arcs to sink \( z \). Further, we define the second part of the second scenario flow \( f^2 \) that sends \( 2n - m \) units along the remaining literal arcs to vertex \( t \). The flow is forwarded to sink \( z \) such that we set

\[
f^2(a) = 1 \text{ for all } \begin{cases} a \in A(q_i^k) \text{ with } q_i^k = v_1 w_1^{2k-1} w_1^{2k+1} r_{2j-1} r_{2j} z \text{if } x_i \text{ is not designated as a verifying literal}, \\ a \in A(q_i^k) \text{ with } q_i^k = v_1 \bar{w}_1^{2k-1} \bar{w}_1^{2k+1} r_{2j-1} r_{2j} z \text{if } \bar{x}_i \text{ is not designated as a verifying literal}, \\ a \in A(q_\ell) \text{ with } q_\ell = t r_{2j-1} r_{2j} z, \ell' \in \{m + 1, \ldots, 2n\} \text{.} \end{cases}
\]

Finally, one further unit is sent along path \( v_1 w_1^e r_{4n+1} r_{4n+3} z \) and one along path \( v_1 \bar{w}_1^e r_{4n+3} z \). We define zero flow on all arcs on which no flow unit has been sent. We have constructed a feasible robust \( b \)-flow \( f = (f^1, f^2) \) for \( \text{ROBT}^\equiv \) instance \( \tilde{I} \). Flow \( f \) satisfies the consistent flow constraints as \( x_1, \ldots, x_n \) is a satisfying truth assignment. Thus, flows \( f^1 \) and \( f^2 \) send each one or zero units (zero or one unit) along fixed arc \((w_i^e, w_i^{e+1}) ((\bar{w}_i^e, \bar{w}_i^{e+1})) \) for \( i \in [n], k \in \{1, 3\} \) and one unit along fixed arc \((r_\ell, r_{\ell+1}) \) for \( \ell' \in \{1, 3, \ldots, 2(2n + 2) - 1\} \).

Conversely, let \( f = (f^1, f^2) \) be a feasible robust \( b \)-flow for \( \text{ROBT}^\equiv \) instance \( \tilde{I} \). Flows \( f^1 \) and \( f^2 \) send one and \( 2n + 2 \) units from source \( v_1 \) to sink \( z \), respectively. By construction of the network, the only option to reach the sink requires the usage of at least two fixed arcs, namely one literal arc and one fixed arc of path \( \tilde{p} \) (except for the two paths \( v_1 w_1^e r_{4n+1} r_{4n+3} z \) and \( v_1 \bar{w}_1^e r_{4n+3} z \) which include each only one fixed arc of path \( \tilde{p} \)). Due to the integral flow \( f^1 \) which
sends 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 $4n + 2$ fixed arcs to meet the demand of flow $f^2$. To use the required $4n + 2$ fixed arcs in the first scenario, flow $f^1$ sends the unit along either path $p_i$ or $\overline{p}_i$ for all $i \in [n]$ (but due to the integral requirement and the acyclic construction never both simultaneously) and subsequently along path $\overline{p}$. If flow $f^1$ sends the unit along path $p_i$, $i \in [n], we set $x_i = \text{True}$. If flow $f^1$ sends the unit along path $\overline{p}_i$, $i \in [n]$, we set $x_i = \text{False}$. To use the required $4n + 2$ fixed arcs in the second scenario, flow $f^2$ sends one unit via each clause vertex and $2n - m$ units via vertex $i$. Depending on the first scenario flow, flow $f^2$ sends one unit along either path $v_1 w_i w_i^{R+1} u_j r_{2j-1} r_{2j} z$ or $v_1 w_i w_i^{R+1} u_j r_{2j-1} r_{2j} z$, $\ell \in \{1,3\}, i \in [n]$ for all $j \in [m]$ (but never both simultaneously) due to the consistent flow constraints. In the former case, clause $C_j$ is verified due to the previous assignment $x_i = \text{True}$ 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 previous assignment $x_i = \text{False}$ induced by flow $f^1$ and the fact that $\overline{x}_i \in C_j$ holds. Two extra units are sent along paths $v_1 w_i^{R+1} v_{n+1} r_{2j} z$ and $v_1 w_i^{R+1} v_{n+1} r_{2j} z$ which use the last two fixed arcs of path $\overline{p}$. The two extra units sent are needed, otherwise there might exist a feasible robust flow whose second scenario flow sends a unit along path $v_1 w_i^{R+1} v_{n+1} r_{2j} z$ or $v_1 w_i^{R+1} v_{n+1} r_{2j} z$ which in turn allows one unsatisfied clause. Overall, $x_1, \ldots, x_n$ is a satisfying truth assignment for instance $I$. ■

APPENDIX B

Proof of Lemma 2. Assume the statement is false. There exists an optimal robust $b$-flow $f = (f^1, f^2)$ such that for two successive arcs $a_i^\ell$ and $a_{i+1}^\ell$, $i \in [2n - 2]$ the following holds true

$$f^1(a_i^\ell) = f^2(a_i^\ell) < f^1(a_{i+1}^\ell) = f^2(a_{i+1}^\ell). \quad (B1)$$

For every $i' \in [2n - 2]$ and $\lambda \in \Lambda$ it holds

$$f^\lambda(a_i^{\lambda}) + f^\lambda(a_{i+1}^{\lambda}) = f^\lambda(a_i^{\lambda+1}) + f^\lambda(a_{i+1}^{\lambda+1}) + f^\lambda(a_{i+1}^{\lambda+1}) \quad (B2)$$

due to the flow balance constraints. Transforming equation (B2) and using the assumption (B1), for $\lambda \in \Lambda$ we obtain

$$f^\lambda(a_i^{\lambda}) > f^\lambda(a_{i+1}^{\lambda}) - f^\lambda(a_{i+1}^{\lambda+1}) =: \beta > 0.$$ 

Let $\tilde{f} = (\tilde{f}^1, \tilde{f}^2)$ be the robust $b$-flow which results by redirecting $\beta$ units of flow $f^\lambda$, $\lambda \in \Lambda$ from arc $a_i^{\lambda}$ to arc $a_i^{\lambda+1}$, that is,

$$\tilde{f}^\lambda(a) = \begin{cases} f^\lambda(a) + \beta & \text{for arc } a = a_i^{\lambda}, \\ f^\lambda(a) - \beta & \text{for arc } a = a_{i+1}^{\lambda}, \\ f^\lambda(a) & \text{otherwise.} \end{cases}$$

An example is visualized in Figure B1. As $c(a_i^{\lambda}) > c(a_i^{\lambda+1})$ holds for all $i' \in [2n - 1]$, we obtain a lower bound on the cost of flow $f^\lambda$, $\lambda \in \Lambda$ as follows

$$c(\tilde{f}^\lambda) = \sum_{a \in A_\lambda} c(a) \cdot f^\lambda(a)$$

$$= \sum_{a \in A_\lambda \setminus \{a_i^{\lambda}, a_{i+1}^{\lambda}\}} c(a) \cdot f^\lambda(a) + c(a_i^{\lambda}) \cdot f^\lambda(a_i^{\lambda}) + c(a_{i+1}^{\lambda}) \cdot f^\lambda(a_{i+1}^{\lambda})$$

FIGURE A1 Construction of ROBT instance $\tilde{I}$. 

APPENDIX B

Proof of Lemma 2. Assume the statement is false. There exists an optimal robust $b$-flow $f = (f^1, f^2)$ such that for two successive arcs $a_i^\ell$ and $a_{i+1}^\ell$, $i \in [2n - 2]$ the following holds true

$$f^1(a_i^\ell) = f^2(a_i^\ell) < f^1(a_{i+1}^\ell) = f^2(a_{i+1}^\ell). \quad (B1)$$

For every $i' \in [2n - 2]$ and $\lambda \in \Lambda$ it holds

$$f^\lambda(a_i^{\lambda}) + f^\lambda(a_{i+1}^{\lambda}) = f^\lambda(a_i^{\lambda+1}) + f^\lambda(a_{i+1}^{\lambda+1}) + f^\lambda(a_{i+1}^{\lambda+1}) \quad (B2)$$

due to the flow balance constraints. Transforming equation (B2) and using the assumption (B1), for $\lambda \in \Lambda$ we obtain

$$f^\lambda(a_i^{\lambda}) > f^\lambda(a_{i+1}^{\lambda}) - f^\lambda(a_{i+1}^{\lambda+1}) =: \beta > 0.$$ 

Let $\tilde{f} = (\tilde{f}^1, \tilde{f}^2)$ be the robust $b$-flow which results by redirecting $\beta$ units of flow $f^\lambda$, $\lambda \in \Lambda$ from arc $a_i^{\lambda}$ to arc $a_i^{\lambda+1}$, that is,

$$\tilde{f}^\lambda(a) = \begin{cases} f^\lambda(a) + \beta & \text{for arc } a = a_i^{\lambda}, \\ f^\lambda(a) - \beta & \text{for arc } a = a_{i+1}^{\lambda}, \\ f^\lambda(a) & \text{otherwise.} \end{cases}$$

An example is visualized in Figure B1. As $c(a_i^{\lambda}) > c(a_i^{\lambda+1})$ holds for all $i' \in [2n - 1]$, we obtain a lower bound on the cost of flow $f^\lambda$, $\lambda \in \Lambda$ as follows

$$c(\tilde{f}^\lambda) = \sum_{a \in A_\lambda} c(a) \cdot f^\lambda(a)$$

$$= \sum_{a \in A_\lambda \setminus \{a_i^{\lambda}, a_{i+1}^{\lambda}\}} c(a) \cdot f^\lambda(a) + c(a_i^{\lambda}) \cdot f^\lambda(a_i^{\lambda}) + c(a_{i+1}^{\lambda}) \cdot f^\lambda(a_{i+1}^{\lambda})$$
Proof of Lemma 3. Before assuming the statement is false, we consider two robust b-flows $f = (f^1, f^2)$ and $\tilde{f} = (\tilde{f}^1, \tilde{f}^2)$ for instance $I_n$ given as follows.

For every $i \in [n]$, flows $f^1$ and $f^2$ send one unit along paths $p_{2i}^1$ and $p_{2i}^2$, respectively. The cost of flow $f$ is $c(f) = \max\{c(f^1), c(f^2)\}$ with

$$c(f^1) = \sum_{i=1}^{n} c(p_{2i}^1) = \sum_{i=1}^{n} (nw(2^{n-i} + 2^{n-1} - 1) + s_{2i}) = nw \sum_{i=1}^{n} 2^{n-i} + n^2w(2^{n-1} - 1) + \sum_{i=1}^{n} s_{2i}$$

and

$$c(f^2) = \sum_{i=1}^{n} c(p_{2i}^2) = \sum_{i=1}^{n} (nw(2^i - 2)2^{n-i-1} + s_{2i-1} + Fw) = nw \sum_{i=1}^{n} ((2^i - 2)2^{n-i-1}) + \sum_{i=1}^{n} s_{2i-1} + nw(2^{n+1} - n - 2)$$

$$= n^2w2^{n-1} - nw(2^n - 1) + \sum_{i=1}^{n} s_{2i-1} + nw(2 \cdot 2^n - n - 2) = nw(2^n - 1) + n^2w(2^{n-1} - 1) + \sum_{i=1}^{n} s_{2i-1}.$$
On the fixed arcs, flows $f^a$ and $\tilde{f}^a$ with $\lambda \in \Lambda$ cause cost amounting to
\[
\sum_{a \in A^{\text{fix}_n}} c(a) \cdot f^a(a) = \sum_{i=1}^{n} c(a^*_{i-1}) \cdot (n-i+1) + \sum_{i=1}^{n-1} c(a^*_i) \cdot (n-i) = 0 + nw(2^{n-1}n - 2^n + 1) = c^{\text{fix}_n},
\]
\[
= \sum_{i=1}^{n} c(a^*_{i-1}) \cdot (n-i) + \sum_{i=1}^{n-1} c(a^*_i) \cdot (n-i) = \sum_{a \in A^{\text{fix}_n}} c(a) \cdot \tilde{f}^a(a).
\]

Now assume there exists an optimal robust $b$-flow $\tilde{f} = (\tilde{f}^1, \tilde{f}^2)$ whose cost incurred on the fixed arcs, denoted by $c^{\text{fix}}_{\tilde{f}}$, does not amount to $c^{\text{fix}_n}$, that is, $c^{\text{fix}}_{\tilde{f}} = \sum_{i=1}^{2n-1} c(a^*_i) \cdot \tilde{f}^1(a^*_i) = \sum_{i=1}^{n-1} c(a^*_i) \cdot \tilde{f}^1(a^*_i) \neq c^{\text{fix}_n}$. Firstly, assume that $\tilde{f} = (\tilde{f}^1, \tilde{f}^2)$ is given such that the cost incurred on the fixed arcs $c^{\text{fix}}_{\tilde{f}}$ is less than $c^{\text{fix}_n}$, that is, $c^{\text{fix}}_{\tilde{f}} < c^{\text{fix}_n}$. As the costs of the fixed arcs are $c(a^*_{i-1}) = 0$ for $i \in [n]$ and $c(a^*_i) = 2^{n-i-1}nw$ for $i \in [n-1]$, it holds $c^{\text{fix}}_{\tilde{f}} - c^{\text{fix}}_{\tilde{f}} \geq 2^0nw = nw \geq 2w$. In the following, we focus on the first scenario flow $\tilde{f}^1$. Let $I$ denote the set of indices that indicate along which paths $p_i^1$ flow $\tilde{f}^1$ sends units, that is, $I := \{i \in [2n] | \tilde{f}^1(p_i^1) > 0\}$. If a unit is sent along path $p_i^1$, $i \in I$, integer $s_i$ is added to the total cost of flow $\tilde{f}^1$. The integers of all other arc cost add up to zero because of the telescope sum. Accordingly, as $n$ units are sent from the source along paths $p_i^1$, $i \in I$ to the sink, exactly $n$ (not necessarily different) integers are added to the total cost of flow $\tilde{f}^1$. For the first scenario, the minimum cost path is given by $p_i^{2m}$ as $c(a^*_i) \leq c(a^*_i)$ holds for all $i \in [2n-1]$. If flow $\tilde{f}^1$ sends $n$ units along the minimum cost path $p_i^{2m} = sa^*_1\cdots a^*_i\cdots a^*_n$, the cost would be
\[
n \cdot c(p_i^{2m}) = n \left( \sum_{i=1}^{2n-1} c(a^*_i) + c(a^*_i) \right) = n \left( \sum_{i=1}^{n-1} c(a^*_i) + c(a^*_i) \right). \tag{B3}
\]

However, if $I \setminus \{2n\} \neq \emptyset$ holds, flow $\tilde{f}^1$ sends a number of units along different paths $p_i^1 = sa^*_1\cdots a^*_i\cdots a^*_n$ for $i \in I \setminus \{2n\}$ (which use more detour instead of cross arcs). If a detour arc $a^*_i$ is used instead of the corresponding cross arc $a^*_i$ for $i \in [2n-1]$, the cost rises by $c(a^*_i) - c(a^*_i) = s_i - s_{i+1} + c(a^*_i) > 0$. Thus, comparing the cost incurred by sending $n$ units along path $p_i^{2m}$ with the cost incurred by sending $n$ units along different paths $p_i^1$ for $i \in I$, we obtain additional cost of
\[
\sum_{i \in I} \left( s_i - s_{2n} + \sum_{j=1}^{2n-1} c(a^*_j) \right) \cdot \tilde{f}^1(p_i^1) = \sum_{i \in I} (s_i - s_{2n}) \cdot \tilde{f}^1(p_i^1) + \sum_{i=1}^{2n-1} c(a^*_i) \cdot \tilde{f}^1(a^*_i)
\]
\[
= \sum_{i \in I} (s_i - s_{2n}) \cdot \tilde{f}^1(p_i^1) + \sum_{i=1}^{2n-1} c(a^*_i) \cdot \left( n - \tilde{f}^1(a^*_i) \right)
\]
\[
= \sum_{i \in I} (s_i - s_{2n}) \cdot \tilde{f}^1(p_i^1) + n \sum_{i=1}^{2n-1} c(a^*_i) - c^{\text{fix}}_{\tilde{f}}. \tag{B4}
\]

Let $i^*$ be the index of the smallest integer, that is, $i^* = \arg \min_{i \in [2n]} s_i$. Combining expressions (B3) and (B4) and using the fact that $c^{\text{fix}}_{\tilde{f}} - 2w \geq c^{\text{fix}}_{\tilde{f}}$ and $\sum_{i=1}^{n} s_{2i-1} < 2w$ hold, we obtain a lower bound on the cost of flow $\tilde{f}^1$ as follows
\[
\begin{align*}
\tilde{c}(\tilde{f}^1) &= \sum_{i \in I} c(p_i^1) \cdot \tilde{f}^1(p_i^1) \\
&= n \cdot c(p_i^{2m}) + \sum_{i \in I} (s_i - s_{2n}) \cdot \tilde{f}^1(p_i^1) + n \sum_{i=1}^{2n-1} c(a^*_i) - c^{\text{fix}}_{\tilde{f}} \\
&\geq n \left( \sum_{i=1}^{n} c(a^*_i) + c(a^*_i) \right) + n \cdot s_{i^*} - n \cdot s_{2n} + n \sum_{i=1}^{n-1} c(a^*_i) - c^{\text{fix}}_{\tilde{f}} \\
&= n \left( 2 \sum_{i=1}^{n} c(a^*_i) + c(a^*_i) - s_{2n} \right) + n \cdot s_{i^*} - c^{\text{fix}}_{\tilde{f}} + 2w
\end{align*}
\]
To prove Theorem C1, we use the following auxiliary lemma.

Let non-negative variables $x_1, \ldots, x_n$ be given with a non-ascending restriction $x_i \geq x_{i+1}$ for all $i \in [n-2]$. The equation

$$\sum_{i=1}^{n-1} 2^{n-i-1} \cdot x_i = 2^{n-1} n - 2^n + 1$$

has $x_i = n - i$ for all $i \in [n-1]$ as its only solution.

To prove Theorem C1, we use the following auxiliary lemma.

We obtain $c(\bar{f}) = \max \{ c(\bar{f}^1), c(\bar{f}^2) \} \geq c(\bar{f}^1) > c(\bar{f}^1) = c(\bar{f})$, which contradicts the assumption.

Secondly, assume that $\bar{f} = (\bar{f}^1, \bar{f}^2)$ is given such that the cost incurred on the fixed arcs $c(\bar{f})$ is greater than $c^{\text{fix}}$, that is, $c(\bar{f}) > c^{\text{fix}}$. As the costs of the fixed arcs are $c(a_{2i-1}^c) = 0$ for $i \in [n]$ and $c(a_{2i}^c) = 2^{n-i-1}n$ for $i \in [n-1]$, it holds $c(\bar{f}) - c^{\text{fix}} \geq 2^n n w = n w \geq 2 w$. In the following, we focus on the second scenario flow $\bar{f}^2$. Let $i^*$ be the index of the smallest integer, that is, $i^* = \arg\min_{i \in [2, n]} s_i$. We note that at least cost of $n(s_{i^*} + Fw)$ is incurred by the free shortcut arcs in the second scenario. As $c^{\text{fix}} \geq 2w + c(\bar{f})$ and $\sum_{i=1}^{n} s_{i} \leq 2w$ hold, we obtain a lower bound on the cost of flow $\bar{f}^2$ as follows

$$c(\bar{f}^2) \geq n(s_{i^*} + Fw) + c(\bar{f}) \geq x_i \cdot s_i + nFw + 2w + c(\bar{f}) > nFw + \sum_{i=1}^{n} s_{i} = c(\bar{f}).$$

We obtain $c(\bar{f}) = \max \{ c(\bar{f}^1), c(\bar{f}^2) \} \geq c(\bar{f}^1) > c(\bar{f}) = c(\bar{f})$, which contradicts the assumption.

Consequently, the cost incurred by an optimal robust $b$-flow $f = (f^1, f^2)$ on the fixed arcs is $c(\bar{f})$.

**Proof of Lemma 4.** We assume that there exists an optimal robust $b$-flow $f = (f^1, f^2)$ with $f^1(a_{2i}^c) = f^2(a_{2i}^c) \neq n - i$ for at least one fixed arc $a_{2i}^c \in A_n^{\text{fix}}$. We define variables $x_i := f^1(a_{2i}^c) = f^2(a_{2i}^c)$ for all $i \in [n-1]$. On all fixed arcs, the flow is limited by the supply that is sent from the unique source to the single sink, that is, $0 \leq x_i \leq n$ for $i \in [n-1]$. By Lemma 2, it holds $x_i \geq x_{i+1}$ for $i \in [n-2]$. By Lemma 3, the cost incurred on the fixed arcs is $c(\bar{f})$. As $c(\bar{f}(a_{2i-1}^c)) = 0$ holds for arcs $a_{2i-1}^c \in A_n^{\text{fix}}$, only the cost incurred by the arcs $a_{2i}^c \in A_n^{\text{fix}}$ contributes to $c(\bar{f})$.

Consequently, the following must hold true

$$\sum_{i=1}^{n-1} c(a_{2i}^c) \cdot x_i = \sum_{i=1}^{n-1} 2^{n-i-1} n w \cdot x_i = c(\bar{f}) = n w (2^n - 1 - 2^n + 1). \quad (B5)$$

Overall, variables $x_i, i \in [n-1]$ satisfy the requirements of Theorem C1 in Appendix C. We obtain that the only solution to equation (B5) is $x_i = (n - i)$ for all $i \in [n-1]$. Thus, it holds $f^1(a_{2i}^c) = f^2(a_{2i}^c) = x_i = n - i$ for fixed arcs $a_{2i}^c \in A_n^{\text{fix}}$, which contradicts the assumption. By Lemma 2 and due to the flow balance constraints, it follows $f^1(a_{2i-1}^c) = f^2(a_{2i-1}^c) \in \{ n - i, n - i + 1 \}$ for fixed arcs $a_{2i-1}^c \in A_n^{\text{fix}}$. □

**APPENDIX C**

**Theorem C1.** Let non-negative variables $x_1, \ldots, x_{n-1} \in [0, \ldots, n]$ be given with a non-ascending restriction $x_i \geq x_{i+1}$ for all $i \in [n-2]$. The equation

$$\sum_{i=1}^{n-1} 2^{n-i-1} \cdot x_i = 2^{n-1} n - 2^n + 1$$

has $x_i = n - i$ for all $i \in [n-1]$ as its only solution.

To prove Theorem C1, we use the following auxiliary lemma.
**Lemma C2.** The following statements are equivalent.

(S1) Let non-negative variables \( x_1, \ldots, x_{n-1} \in \{0, \ldots, n\} \) be given with a non-ascending restriction \( x_i \geq x_{i+1} \) for all \( i \in [n-2] \). The equation

\[
\sum_{i=1}^{n-1} 2^{n-i} \cdot x_i = 2^{n-1} n - 2^n + 1
\]

has \( x_i = n - i \) for all \( i \in [n-1] \) as its only solution.

(S2) Let non-negative variables \( \bar{x}_0, \ldots, \bar{x}_{n-2} \in \{0, \ldots, n\} \) be given with a non-descending restriction \( \bar{x}_i \leq \bar{x}_{i+1} \) for all \( i \in \{0, \ldots, n-3\} \). The equation

\[
\sum_{i=0}^{n-2} 2^i \cdot \bar{x}_i = 2^{n-1} n - 2^n + 1
\]

has \( \bar{x}_i = i + 1 \) for all \( i \in \{0, \ldots, n-2\} \) as its only solution.

(S3) Let non-negative variables \( y_0, \ldots, y_{n-2} \in \{0, \ldots, n\} \) be given with \( \sum_{i=0}^{n-2} y_i \leq n \). The equation

\[
2^{n-1} \sum_{i=0}^{n-2} y_i - \sum_{i=0}^{n-2} 2^i y_i = 2^{n-1} n - 2^n + 1
\]

has \( y_i = 1 \) for all \( i \in \{0, \ldots, n-2\} \) as its only solution.

**Proof.** Firstly, by shifting the index of the left hand side of equation (E1) and summing up the other way around, we obtain

\[
\sum_{i=0}^{n-1} 2^{n-i} \cdot x_i = \sum_{i=0}^{n-2} 2^{n-i-2} \cdot x_i' = \sum_{i=0}^{n-2} 2^i \cdot \bar{x}_i
\]

with \( x_i' := x_{i+1} \) for all \( i \in \{0, \ldots, n-2\} \) and \( \bar{x}_i := x_i' - x_{i-1} \) for all \( i \in \{0, \ldots, n-2\} \). If \( x_i \geq x_{i+1} \) is true for all \( i \in [n-2] \), it holds \( \bar{x}_i \leq \bar{x}_{i+1} \) for all \( i \in \{0, \ldots, n-3\} \). The assignment \( y_i = n - i, i \in [n-1] \) is the only solution to equation (E1) if and only if \( \bar{x}_i = i + 1, i \in \{0, \ldots, n-2\} \) is the only solution to equation (E2). Consequently, statement (S1) is equivalent to statement (S2).

We define new variables \( y_0 := \bar{x}_0 \) and \( y_i := \bar{x}_i - \bar{x}_{i-1} \) for all \( i \in [n-2] \). Because of \( \bar{x}_0 \geq 0 \) and \( \bar{x}_i \leq \bar{x}_{i+1} \) for \( i \in \{0, \ldots, n-3\} \), it holds \( y_i \geq 0 \) for all \( i \in \{0, \ldots, n-2\} \). Furthermore, for all \( i \in \{0, \ldots, n-2\} \) it holds \( \bar{x}_i = \sum_{j=0}^{i} y_j \) and as \( \bar{x}_{n-2} \in \{0, \ldots, n\} \) we obtain \( \sum_{i=0}^{n-2} y_i \leq n \). Using this, we obtain

\[
\sum_{i=0}^{n-2} 2^i \cdot \bar{x}_i = \sum_{i=0}^{n-2} 2^i \cdot \left( \sum_{j=0}^{i} y_j \right) = \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} \begin{pmatrix} i \\ j \end{pmatrix} 2^j y_j = \sum_{j=0}^{n-2} \left( 2^{n-1} - 2^j \right) y_j = 2^{n-1} \sum_{j=0}^{n-2} y_j - \sum_{j=0}^{n-2} 2^j y_j.
\]

The assignment \( \bar{x}_i = i + 1, i \in \{0, \ldots, n-2\} \) is the only solution to equation (E2) if and only if \( y_0 = \bar{x}_0 = 1 \) and \( y_i = \bar{x}_i - \bar{x}_{i-1} = i + 1 - i = 1 \) is the only solution to equation (E3). Consequently, statement (S2) is equivalent to statement (S3).

In the following, we focus on statement (S3) of Lemma C2 to prove Theorem C1. For the proof, we use Algorithm C.1 to transform solutions to equation

\[
\sum_{i=0}^{n-2} 2^i y_i = 2^n - 1.
\]

More precisely, for a given parameter \( \ell \in \mathbb{Z}_{\geq 2} \) the algorithm transforms an arbitrary solution \( y = y^f \) to equation (C1) after \( j - 1 \) steps to a different solution \( y^f' \) to equation (C1) with \( y^f'_{i+1} = \ell \) or \( y^f'_{i+1} \in \{0, 1\} \) for \( i \in \{0, \ldots, n-3\} \) and \( y^f'_{n-2} \in \{0, 1, \ldots, \ell-1\} \) such that the sum of the entries is reduced by \( j - 1 \), that is,

\[
\sum_{i=0}^{n-2} y^f'_i = \sum_{i=0}^{n-2} y^f_i - (j - 1).
\]

The algorithm is used in the next lemma to show that a solution assumed to exist does not satisfy equation (E3).
Algorithm C.1.

1. **Require**: solution \( y = (y_0, \ldots, y_{n-2}) \in \mathbb{Z}_{\geq 0}^{n-1} \) and parameter \( \ell' = n \in \mathbb{Z}_{\geq 2} \)
2. Set \( j = 1 \) and \( y_1 := y \)
3. **procedure**
4. while \( \exists k \in \{0, \ldots, n-3\} : y_k \geq 2 \) do
5. Let \( k \) be minimal
6. Set \( y_{j+1} = \begin{cases} y_k - 2 & \text{for } i = k \\ y_{j+1} + 1 & \text{for } i = k + 1 \\ y_j & \text{otherwise} \end{cases} \)
7. Set \( j = j + 1 \)
8. if \( y_{n-2} = \ell' \) then
9. break
10. return \( y' = (y_0', \ldots, y_{n-2}') \)

**Lemma C3.** Let \( y = (y_0, \ldots, y_{n-2}) \in \{0, \ldots, n\}^{n-1} \) with \( \sum_{i=0}^{n-2} y_i \leq n \) be a solution to the equation

\[
2^{n-1} \sum_{i=0}^{n-2} y_i - \sum_{i=0}^{n-2} y_i^2 = 2^{n-1} n - 2^n + 1. \tag{E3}
\]

Then, it holds \( \sum_{i=0}^{n-2} y_i = n - 1 \).

**Proof.** Assume the statement is false, then it either holds \( \sum_{i=0}^{n-2} y_i \leq n - 2 \) or \( \sum_{i=0}^{n-2} y_i = n \). In the first case, we obtain

\[
2^{n-1} n - 2^n + 1 = 2^{n-1} \sum_{i=0}^{n-2} y_i - \sum_{i=0}^{n-2} y_i^2 \leq 2^{n-1} (n - 2) - \sum_{i=0}^{n-2} y_i^2 < 2^{n-1} (n - 2) + 1 = 2^{n-1} n - 2^n + 1,
\]

which is a contradiction. In the second case, the equation (E3) reduces to the following equivalent equation

\[
2^{n-1} n - \sum_{i=0}^{n-2} y_i = 2^{n-1} n - 2^n + 1 \iff \sum_{i=0}^{n-2} y_i = 2^n - 1. \tag{C2}
\]

For a solution \( y \in \mathbb{Z}_{\geq 0}^{n-1} \) to equation (C2) there exists a \( k \in \{0, \ldots, n-2\} \) such that \( y_k \geq 2 \) holds by the pigeonhole principle. Let \( k \) be minimal. If \( k = n - 2 \), we distinguish between the following two cases.

1. **Case** \( y_{n-2} \geq 4 \): We obtain

\[
2^n - 1 = \sum_{i=0}^{n-2} y_i^2 \geq 2^{n-2} y_{n-2} = 2^{n-2} \cdot 4 = 2^n > 2^n - 1,
\]

which is a contradiction.

2. **Case** \( y_{n-2} \in \{2, 3\} \): We note that \( y' = (y_0', \ldots, y_{n-2}') = (1, \ldots, 1, 3) \) satisfies \( \sum_{i=0}^{n-2} y_i' = 2^n - 1 \) but it holds \( \sum_{i=0}^{n-2} y_i' = n - 2 + 3 = n + 1 \). As \( k = n - 2 \) is minimal, solution \( y \) consists of \( y_i \in \{0, 1\} \) for \( i \in \{0, \ldots, n-3\} \) and \( y_{n-2} \in \{2, 3\} \). As \( \sum_{i=0}^{n-2} y_i = n < n + 1 = \sum_{i=0}^{n-2} y_i' \) is true, it holds \( y_i = 0 \) for at least one \( i \in \{0, \ldots, n-3\} \) or it holds \( y_{n-2} = 2 \). In both cases, we obtain

\[
2^n - 1 = \sum_{i=0}^{n-2} y_i^2 < \sum_{i=0}^{n-2} y_i' = 2^n - 1,
\]

which is a contradiction.

If \( 0 \leq k < n - 2 \), we apply Algorithm C.1 on solution \( y \) for \( \ell' = 4 \). After \( j - 1 > 0 \) steps, the algorithm terminates and provides a solution \( y' \in \mathbb{Z}_{\geq 0}^{n-1} \) with \( y'_{n-2} = \ell' \) or \( y_j' \in \{0, 1\} \) for \( i \in \{0, \ldots, n-3\} \) and \( y_{n-2}' \in \{0, 1, \ldots, \ell' - 1\} \) such that the following holds

\[
\sum_{i=0}^{n-2} y'_i = \sum_{i=0}^{n-2} y_i - (j - 1) = n - j + 1.
\]
If $y_{n-2} = \ell = 4$, we obtain analogously to Case 1

$$2^n - 1 = \sum_{i=0}^{n-2} 2^i y_i = \sum_{i=0}^{n-2} 2^i y'_i \geq 2^{n-2} y_{n-2} = 2^{n-2} \cdot 4 = 2^n > 2^n - 1,$$

which is a contradiction. Otherwise, solution $y'$ consists of $y'_i \in \{0, 1\}$ for $i \in \{0, \ldots, n-3\}$ and $y'_{n-2} \in \{0, 1, 2, 3\}$. Analogues to Case 2, we obtain

$$2^n - 1 = \sum_{i=0}^{n-2} 2^i y_i = \sum_{i=0}^{n-2} 2^i y'_i < \sum_{i=0}^{n-2} 2^i y'_i = 2^n - 1,$$

which is a contradiction. Overall, we have proven $\sum_{i=0}^{n-2} y_i = n - 1$.

**Lemma C3.** Let non-negative variables $y_0, \ldots, y_{n-2} \in \{0, \ldots, n\}$ be given with $\sum_{i=0}^{n-2} y_i = n - 1$. The equation

$$2^{n-1} \sum_{i=0}^{n-2} y_i - \sum_{i=0}^{n-2} 2^i y_i = 2^{n-1} n - 2^n + 1 \quad (E3)$$

has $y_i = 1$ for all $i \in \{0, \ldots, n-2\}$ as its only solution.

**Proof.** As $\sum_{i=0}^{n-2} y_i = n - 1$ holds, the equation (E3) reduces to the following equivalent equation

$$2^{n-1}(n - 1) - \sum_{i=0}^{n-2} 2^i y_i = 2^{n-1} n - 2^n + 1 \iff \sum_{i=0}^{n-2} 2^i y_i = 2^{n-1} - 1. \quad (C3)$$

Assume now the statement is false, that is, there exists a solution $y^d = (y^d_0, \ldots, y^d_{n-2}) \in \{0, \ldots, n\}^{n-1}$ to equation (C3) with $\sum_{i=0}^{n-2} y^d_i = n - 1$ where $y^d \neq y^r := (1, \ldots, 1)$ holds. There exists a $k \in \{0, \ldots, n-2\}$ such that $y^d_k \geq 2$ holds by the pigeonhole principle. Let $k$ be minimal. If $k = n - 2$, we obtain

$$2^n - 1 = \sum_{i=0}^{n-2} 2^i y'_i \geq 2^{n-2} y_{n-2} \geq 2^{n-2} \cdot 2 = 2^{n-1} > 2^n - 1,$$

which is a contradiction. If $0 \leq k < n - 2$, we apply Algorithm C.1 on solution $y^d$ for $\ell = 2$. After $j - 1 > 0$ steps, the algorithm terminates and provides a solution $y^f \in \mathbb{Z}^{n-1}_{\geq 0}$ with $y^f_{n-2} = \ell$ or $y^f_i \in \{0, 1\}$ for $i \in \{0, \ldots, n-3\}$ and $y^f_{n-2} \in \{0, 1, \ldots, \ell - 1\}$ such that the following holds

$$\sum_{i=0}^{n-2} y^f_i = \sum_{i=0}^{n-2} y^d_i - (j - 1) = n - j.$$

If $y^f_{n-2} = \ell = 2$, we obtain a contradiction analogues to expression (C4). Otherwise, solution $y^f \in \mathbb{Z}^{n-1}_{\geq 0}$ consists of $y^f_i \in \{0, 1\}$ for all $i \in \{0, \ldots, n-2\}$. Furthermore, as at least one step of the algorithm has been performed, it holds $y^f_i = 0$ for at least one $i \in \{0, \ldots, n-2\}$. We obtain

$$2^n - 1 = \sum_{i=0}^{n-2} 2^i y^f_i = \sum_{i=0}^{n-2} 2^i y^f_i < \sum_{i=0}^{n-2} 2^i y^f_i = 2^n - 1,$$

which is a contradiction.

**APPENDIX D**

**Lemma D1.** Let $G$ be an SP digraph and $v, w \in V(G)$ be two specified vertices such that there exists a $(v, w)$-path. Let $G_{vw}$ be the subgraph spanned by vertices $v, w$. Subgraph $G_{vw}$ is an SP digraph itself with origin $v$ and target $w$.

**Proof.** We prove the correctness of the statement by induction on the number of the digraph’s arcs $m \coloneqq |A(G)|$. For the beginning, if we consider an SP digraph consisting of only one arc, the statement is readily apparent. In the next step, we assume the statement holds true for all SP digraphs consisting of at most $m$ arcs. In the following, we prove the statement for an SP digraph with $m + 1$ arcs. We distinguish whether SP digraph $G$ is composed serially or in parallel.
Firstly, we assume that $G$ is a series composition of SP digraphs $G_1$ and $G_2$. The target of SP digraph $G_1$ and the origin of SP digraph $G_2$ are contracted to one vertex, denoted by $z$. Let $v, w \in V(G)$ be two specified vertices such that there exists a $(v, w)$-path. If $v, w \in V(G_1)$ or $v, w \in V(G_2)$ holds, the statement results by the induction hypothesis. Otherwise, it holds $v \in V(G_1)$ and $w \in V(G_2)$ due to the $(v, w)$-path. We separately consider SP digraphs $G_1$ and $G_2$ with specified vertices $v, z \in V(G_1)$ and $z, w \in V(G_2)$, respectively. Clearly, $(v, z)$- and $(z, w)$-paths exist, as the $(v, w)$-path must contain the connecting vertex $z$. By the induction hypothesis, we obtain that the spanned subgraphs $G_{vz}$ and $G_{zw}$ are SP digraphs themselves. If we serially compose subgraphs $G_{vz}$ and $G_{zw}$ at vertex $z$, we obtain subgraph $G_{vw}$ spanned by vertices $v, w$. Subgraph $G_{vw}$ is an SP digraph as it is a series composition of two SP digraphs.

Secondly, we assume that $G$ is a parallel composition of SP digraphs $G_1$ and $G_2$. Let $v, w \in V(G)$ be two specified vertices such that there exists a $(v, w)$-path. It holds $v, w \in V(G_1)$ or $v, w \in V(G_2)$ and thus the statement results by the induction hypothesis.