Robust and Adaptive Network Flows

Dimitris Bertsimas, Ebrahim Nasrabadi
Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 [dbertsim@mit.edu, nasrabad@mit.edu]

Sebastian Stiller
Institut für Mathematik, Technische Universität Berlin, 10623 Berlin, Germany, sebastian.stiller@tu-berlin.de

We study network flow problems in an uncertain environment from the viewpoint of robust optimization. In contrast to previous work, we consider the case that the network parameters (e.g., capacities) are known and deterministic, but the network structure (e.g., nodes and arcs) is subject to uncertainty. In this paper, we study the robust and adaptive versions of the maximum flow problem and minimum cut problems in networks with node and arc failures, and establish structural and computational results. The adaptive two-stage model adjusts the solution after the realization of the failures in the network. This leads to a more flexible model and yields less conservative solutions compared to the robust model.

We show that the robust maximum flow problem can be solved in polynomial time, but the robust minimum cut problem is NP-hard. We also prove that the adaptive versions are NP-hard. We further characterize the adaptive model as a two-person zero-sum game and prove the existence of an equilibrium in such games.

Moreover, we consider a path-based formulation of flows in contrast to the more commonly used arc-based version of flows. This leads to a different model of robustness for maximum flows. We analyze this problem as well and develop a simple linear optimization model to obtain approximate solutions. Furthermore, we introduce the concept of adaptive maximum flows over time in networks with transit times on the arcs. Unlike the deterministic case, we show that this problem is NP-hard on series-parallel graphs even for the case that only one arc is allowed to fail. Finally, we propose heuristics based on linear optimization models that exhibit strong computational performance for large-scale instances.

Subject classifications: network flows; robust optimization; game theory; flows over time.

Area of review: Optimization.

History: Received May 2012; revisions received December 2012, February 2013; accepted March 2013.

1. Introduction

Network flow problems form an important class of optimization problems and are a central topic in operations research, computer science, and combinatorial optimization. They have been investigated and expanded by many researchers from various points of view. Numerous efficient algorithms for solving the maximum flow problem, the shortest-path problem, and the minimum cost flow problem exist. A comprehensive discussion of theory, algorithms, and applications of network flow problems can be found in Ahuja et al. (1993).

In most real-world applications such as production systems, communication networks, financial flows, and pipeline systems for transporting, data are subject to uncertainty. In particular, a crucial characteristic of a wide range of network flow applications is that the network structure is itself uncertain; for example, in wireless communication networks, arcs break down and nodes appear and disappear. This characteristic is not captured by classical network flow models where parameters must be precise and the network structure must be known beforehand.

We consider network flows in the framework of Robust Optimization (see, e.g., Ben-Tal and Nemirovski 1998, 1999a, b; Bertsimas and Sim 2003, 2004; El-Ghaoui and Lebret 1997, El-Ghaoui et al. 1998). In this framework, the input data of a problem can vary within an uncertainty set. A solution is feasible if it is feasible for every realization of the data within the uncertainty set. Different uncertainty sets lead to different modeling power, different levels of conservatism, and differences in computational tractability. Although all robust optimization models in the literature follow this worst-case approach, the main distinguishing feature of the models for robust optimization is the type of uncertainty set considered. For example, Ben-Tal and Nemirovski (1998, 1999a, b) and El-Ghaoui et al. (1998) work with ellipsoidal uncertainty sets, whereas Bertsimas and Sim (2003, 2004) consider polyhedral uncertainty sets. They show that a robust solution can be obtained by solving a problem that is similar to the nominal problem in complexity, type, and size.

Bertsimas and Sim (2003) study robust optimization for discrete optimization and network flow problems. They show that the robust counterpart of a polynomially solvable 0-1 discrete optimization problem (e.g., robust matching, spanning tree, shortest path) remains polynomially solvable. They also show that the robust counterpart of an NP-hard α-approximable 0-1 discrete optimization problem remains α-approximable.
In recent years, adaptive optimization models have been proposed to address multistage decision problems under uncertainty. In particular, Ben-Tal et al. (2004) propose a two-stage adaptive approach in which the decision maker makes two sets of decisions: one set before the uncertainty being realized and one set subsequently. They show that the adaptive counterpart of a linear optimization problem with right-hand-side uncertainty is, in general, NP-hard. We refer to Ben-Tal et al. (2004), Bertsimas and Goyal (2010), and Bertsimas et al. (2011) for an overview on approximation algorithms for this model. Liebchen et al. (2009) present a similar concept of adaptive optimization, the so-called recoverable robustness. Here, the decision maker looks for a solution together with a recovery algorithm, which recovers the solution to a feasible solution in every scenario.

Adaptive optimization approaches have been applied for solving various network flow problems (see, e.g., Atamtürk and Zhang 2007, Chekuri et al. 2007, Mattia 2013, Ordóñez and Zhao 2007, Poss and Raack 2011). In particular, Atamtürk and Zhang (2007) present an adaptive optimization approach for network flow and network design problems with demand uncertainty. They give an explicit characterization of the first-stage decisions with an exponential number of constraints, and prove that the corresponding separation problem is NP-hard even for a network flow problem on a bipartite graph. They show, however, that if the second-stage network topology is totally ordered or an arborescence, then the separation problem is tractable. Poss and Raack (2011) introduce the concept of affine routing for adaptive network design with polyhedral demand uncertainty. They present a theoretical and empirical study of the resulting model.

Unlike these works on demand robustness, our work considers network flow problems with arc failures. This is closely related to the network interdiction problem, which has been studied in the literature (see, e.g., Cormican et al. 1998, Altner et al. 2010, Roysset and Wood 2007, Washburn and Wood 1995, Wood 1993). The network interdiction problem asks to find a certain number of arcs whose removal from the network minimizes the maximum amount of flow that one can send through the network. This problem appears in a wide variety of areas such as drug interdiction, military planning, protecting electric power grids against terrorist attacks, and hospital infection control. Wollmer (1964) proposes a polynomial time algorithm for solving the network interdiction problem on planar graphs. On general networks, Wood (1993) shows that the problem is strongly NP-complete. Minoux (2009) examines the network interdiction problem from a different perspective, in which the network structure is fixed, but, rather, the arc capacities are uncertain. In particular, he assumes that the uncertainty set is the solution set of an associated (continuous) knapsack problem and shows that the resulting problem is NP-hard.

In the network interdiction problem, the goal is to determine the worst-case scenario, assuming the decision maker is in a position to act after the realization of uncertainty, i.e., to implement a maximum flow when the set of failed arcs is realized. However, in many applications, one has to make a decision before the realization of the uncertain data. In particular, one is interested in those solutions that are robust against any possible scenario. Many practical situations, particularly in network optimization, naturally give rise to uncertain linear optimization models where the parameters lie in discrete uncertainty sets. We next present a real-world application to motivate the study of network flow problems with node/arc failures.

We consider a problem of pumping crude oil around a crude oil distribution system. A crude oil distribution system is the essential part of a crude oil supply chain that links the upstream and downstream functions. Obviously, the management of this system has a critical role in the performance of the crude oil supply chain.

A typical large oil company operates more than 10 refineries that process several million barrels of crude oil every day. There are several methods that are used to transport crude oil: pipelines, tank trucks, railroad tank cars, barges, and tankers. Pipelines are very economical, especially when large amounts of crude oil have to be pumped over long distances. We consider a crude oil network linking a number of production units to consumption centers (refineries and export terminals) by pipelines. In addition, there are intermediate pump stations in long-distance networks to keep the crude oil moving smoothly through pipelines. Traditionally, this system is mostly managed manually without much assistance from scientific tools. The traditional (manual) methods do not take into account that pipelines or pump stations may be out of service. For example, during a war this can become a crucial problem because oil facilities across the country are suddenly under attack. The situation is aggravated by the fact that whenever pipelines are bombed or attacked, most other industries are affected as well. Hence, it is important to manage the crude oil transportation system in such a way that it is capable of coping with such situations and to reduce the amount of shortage of crude oil at consumption centers as much as possible.

The challenge is to construct the pipelines such that (a) they operate economically under normal conditions and (b) they guarantee a high residual transportation capacity even after the worst possible attack. We model the requirement (a) as the maximum feasible flow value in the worst-case scenario. We do this in two ways. First, we look at the flow that will reach the sink immediately after an attack, i.e., without any measures taken to reduce the effects of the attack. Secondly, in the main part of the paper we consider the maximum flow that can be routed through the network after the attack, i.e., the flow that can pass the network, if one has time to reroute but not to rebuild. The requirement (a) that normal operations must be economical can be
modeled in different ways, e.g., by a cost for the sum of the capacities of the installed pipelines. However, such costs imply a fixed ratio between the nominal operation costs and the importance given to the disruption case. This ratio is difficult to fix if the probability for a disruption cannot be easily quantified. Thus, we use a different rationale in this paper. We say normal operation is economical if it is a flow, that is, if the installed capacity of every segment is fully used in normal operations. There are also technical reasons (e.g., pressure in the tube) why each segment of a pipeline should operate at its capacity. This requirement may be dropped in the emergency operations after an attack, but not for long-term, normal use.

The requirement that the structure for normal operations has to be a flow also appears in other applications. A logistic network consists of sources, sinks, and several hubs connected by different means of transportation, e.g., trucks or railways. The sinks are either clients or production facilities. On the one hand, in normal operations trucks should run full or almost full, i.e., the chosen network should be a flow. On the other hand, even in a scenario of failing hubs or transportation links, a certain amount of goods must reach the destination; otherwise, the production might come to a hold, or sales are affected substantially.

Note that many research models use network flows as an ingredient of more complex models. Logistic networks, for example, usually require more complex models based on flows. Traditional network flows tend to accumulate the flow on few paths and arcs. Apparently, this is not very robust against arc failure. This work presents alternative models that are still flows but are more robust, so they are candidates to replace the classical flows in flow-based models that require robustness.

Our contribution. The contributions of this paper are as follows:

1. We introduce the notions of robust and adaptive network flows to address real-world applications, in which arcs can fail and the overall network structure is uncertain.
2. We provide insights into the mathematical structure, complexity, approximability, and computability of the robust and adaptive versions of static and dynamic maximum flow problems.

Structure of the Paper. In §2, we study robust flows that remain feasible under all possible failures of arcs and nodes. Traditionally, feasibility of a network flow requires obeying strict flow conservation at each node; that is, the sum of inflows must be equal to the sum of outflows. Under this condition, the failure of any arc with positive flow makes every flow infeasible. Therefore, in our formulation of the robust maximum flow problem, we relax strict flow conservation at each node to weak flow conservation. Hence, the sum of inflows can be more than the sum of outflows, and the sums do not have to be identical. We show that this problem can be formulated as a linear optimization (LO) model, and is consequently solvable in polynomial time. Moreover, we consider the dual of the robust maximum flow problem and introduce the robust minimum cut problem by imposing integrality constraints on the variables. We show that the resulting problem is strongly NP-hard.

A flow that remains feasible without adjustments in every failure scenario may be still a very conservative structure. We present in §3 a more flexible and less conservative model of robustness, the adaptive model. In this model, we allow the flow to adjust after the realization of failures in the network. More precisely, we turn the infeasible flow into a feasible flow by solving a maximum flow problem in the remaining network so that the flow on each arc does not exceed the initial flow. We give a characterization of the adaptive flows and present structural results. We formulate the problem as an LO model with exponentially many variables and constraints. For a small number of arc failures, the LO model has polynomially many variables and constraints, and can be solved in polynomial time. However, in general, we show that the adaptive maximum flow problem is NP-hard by a reduction from the network interdiction problem.

In §3, we also introduce the adaptive minimum cut problem and design a two-person zero-sum game, in which one player chooses a flow and the other player (an adversary) selects a cut to reduce the maximum flow value as much as possible by attacking certain arcs in that cut. We prove the existence of an equilibrium in such games.

There are two approaches to model network flow problems: the arc-based formulations and the path-based formulation, depending on whether we define flows on arcs or on paths and cycles, respectively. In §4, we consider a variant of robust flows in the path-based formulation. This problem is closely related, but not equivalent, to the adaptive model in arc-based formulation. We investigate the relationship between these two approaches and propose an LO-based method to obtain approximate solutions for both models.

In addition to the uncertainty of the network structure, temporal dynamics is an interesting and important feature of network flow applications. To address this aspect, we extend the concept of adaptivity to maximum flows over time in §5. Unlike the maximum flow-over-time problem, we observe that the class of temporally repeated flows need not contain an adaptive maximum flow over time. We further show that the problem of finding an adaptive maximum flow over time is weakly NP-hard.

Section 6 is devoted to computational results. We report extensive results to demonstrate the ability of the proposed LO method to obtain near-optimal solutions for both the robust path-based and adaptive arc-based problems. The final section includes our conclusion.

2. The Robust Maximum Flow Problem

We consider network flows that are robust against the failure of any arc set of bounded cardinality. This is motivated by many practical applications where nodes disappear or
arcs break down. For example, the aforementioned pipeline system can be subject to a natural catastrophe or the attack of a malign adversary.

Traditionally, a feasible flow has to fulfill strict flow conservation at every node. Under this requirement, every flow becomes infeasible after the failure of any flow-carrying arc. Therefore, it is natural to allow the flow to adjust after the failures. We study adaptive flows in the next section. In this section, we take a simpler approach relaxing the strict flow conservation constraints to the weak flow conservation constraints. Note that for classical maximum flows the two are equivalent. In terms of application, weak flow conservation constraints. Note that for classical maximum flows

\[ \sum_{e \in \delta^-(v)} x_e - \sum_{e \in \delta^+(v)} x_e \geq 0. \]

Throughout the paper, we let \( N \) be the node-arc adjacency matrix of the extended network after deleting the row corresponding to node \( s \). Then, the maximum flow problem is formulated as follows:

\[
\begin{align*}
\max & \quad x_{(t,i)} \\
\text{s.t.} & \quad Nx \geq 0, \\
& \quad 0 \leq x \leq u.
\end{align*}
\]

The aim of the problem is to maximize the flow on the arc \((t,s)\), which is equal to the value of the corresponding \(s\)-t-flow in the original network.

**Node failure**: In what follows, we will only consider failure of arcs. However, it might be the case in some applications that nodes fail. Notice that whenever a node fails, the arcs in and out of that node disappear as well. We describe next how the general model with node and arc failures can be transformed into the case where only arcs can fail. To this end, we split each node \( v \) into two nodes \( v' \) and \( v'' \), and add the arc \( e_v := (v',v'') \) (see Figure 1). All outgoing arcs of \( v \) are now leaving node \( v'' \), and those that entered \( v \) now go into \( v' \). One can easily observe that the robust flow problem in network \( G \) with node and arc failures is equivalent to the robust flow problem in the network constructed from \( G \) where only arcs can fail.

In practice, it is unlikely that all arcs fail simultaneously. To control the level of conservatism, we consider an integer parameter \( \Gamma_v \) \((0 \leq \Gamma_v \leq |\delta^-(v)|)\) for each node \( v \in V \setminus \{s\} \). The parameter \( \Gamma_v \) gives an upper bound on the number of incoming arcs into \( v \) that can fail. We use

\[ \Lambda_v := \left\{ \mu = (\mu_e)_{e \in \delta^-(v)} \in [0,1]^{\delta^-(v)} \mid \sum_{e \in \delta^-(v)} \mu_e \leq \Gamma_v \right\} \]

(2)

to denote the set of all possible scenarios at node \( v \). The binary variable \( \mu_e \) indicates whether or not arc \( e \) fails, depending on whether \( \mu_e = 1 \) or \( \mu_e = 0 \), respectively.

We say that a flow \( x \) is robust if it satisfies the weak flow conservation constraint at each node \( v \in V \setminus \{s\} \) under any possible scenario, i.e.,

\[ \sum_{e \in \delta^-(v)} (1-\mu_e) x_e - \sum_{e \in \delta^+(v)} x_e \geq 0 \quad \forall v \in V \setminus \{s\}, \mu \in \Lambda_v. \]
Each node $v \in V$ is split into two nodes, $v'$ and $v''$, and the arc $e_v := (v', v'')$ is added to the network.

Figure 1. Each node $v \in V$ is split into two nodes, $v'$ and $v''$, and the arc $e_v := (v', v'')$ is added to the network.

Note. The robust network flow problem in the original network with arc and node failures is equivalent to the robust flow problem in the constructed network where only arcs can fail.

The value of $x$ is $x_{(s,t)}$, the flow on the artificial arc. We refer to a robust flow with maximum value as a robust maximum flow. We can define the problem as follows:

**Robust Maximum Flow Problem**

Input: A network $G = (V, E)$ with a source $s \in V$, a sink $t \in V$, arc capacities $u_e \in \mathbb{R}_+$ for $e \in E$ such that for each node $v \in \{s\}$ at most $\Gamma$ incoming arcs into $v$ can fail.

Task: Find a robust flow with maximum value.

The problem can be formulated as the following LO model:

$$\text{max } x_{(s,t)}$$

$$\text{s.t. } \sum_{e \in \delta^-(v)} (1 - \mu_e) x_e - \sum_{e \in \delta^+(v)} x_e \geq 0$$

$$\forall v \in V \setminus \{s\}, \mu \in \Delta_n, 0 \leq x \leq u.$$  \hspace{1cm} (4)

This problem has exponentially many constraints. However, we show that the robust maximum flow problem can be turned into a LO model with $2m + n - 1$ variables and $2m + n - 1$ constraints.

**Theorem 1.** The following LO model solves the robust maximum flow problem:

$$\text{max } x_{(s,t)}$$

$$\text{s.t. } \sum_{e \in \delta^-(v)} (x_e - \theta_e) - \sum_{e \in \delta^+(v)} x_e - \Gamma \delta_v \geq 0$$

$$\forall v \in V \setminus \{s\},$$

$$x_e - \theta_e - \delta_v \leq 0 \ \forall e = (v, w) \in E,$$

$$0 \leq x_e \leq u_e \ \forall e \in E,$$

$$\theta_e \geq 0 \ \forall e \in E,$$

$$\delta_v \geq 0 \ \forall v \in V \setminus \{s\}.$$  \hspace{1cm} (5)

Therefore, the robust maximum flow problem can be solved in time polynomial in the size of the input $(G, u, \Gamma)$.

**Proof.** It follows from the definition of robust flows that an $s$-$t$-flow $x$ is robust if and only if

$$\text{min } \mu \in \Delta_n \left\{ \sum_{e \in \delta^-(v)} (1 - \mu_e) x_e - \sum_{e \in \delta^+(v)} x_e \geq 0 \ \forall v \in V \setminus \{s\} \right\},$$

or, equivalently,

$$\sum_{e \in \delta^-(v)} x_e - \sum_{e \in \delta^+(v)} x_e - z_e(x) \geq 0 \ \forall v \in V \setminus \{s\},$$

$$\text{where } z_e(x) \text{ is the sum of the first } \Gamma \text{ biggest arc flow values } x_e \text{ into node } v, \text{ that is, } z_e(x) := \max_{e \in \delta^+(v)} x_e \mu_e$$

$$\text{s.t. } \sum_{e \in \delta^-(v)} x_e \geq z_v(x)^{\ast} \quad \forall v \in V \setminus \{s\},$$

$$0 \leq x_e \leq u_e \quad \forall e \in E,$$

$$\theta_e \geq 0 \quad \forall e \in E,$$

$$\delta_v \geq 0 \quad \forall v \in V \setminus \{s\}.$$  \hspace{1cm} (7)

By considering the dual of Problem (8), we obtain

$$z_v(x) = \min \sum_{e \in \delta^+(v)} \theta_e + \Gamma \delta_v$$

$$\text{s.t. } \theta_e + \delta_v \geq x_e \quad \forall e \in \delta^+(v),$$

$$\theta_e \geq 0 \quad \forall e \in \delta^-(v),$$

$$\delta_v \geq 0.$$  \hspace{1cm} (9)

The result now follows by substituting $z_v(x)$ in constraints (7). $\square$

The weak flow conservation in this model has the following interpretation: The flow is designed to guarantee the outflow at every node $v$ even if $\Gamma$ of the inflow-carrying arcs fail. Therefore, the inflow without failure of arcs must have an excess over the guaranteed outflow, i.e., only weak flow conservation holds.

**The Robust Minimum Cut Problem:** The MaxFlow-MinCut Theorem of Ford and Fulkerson (1962) is one of the cornerstones in network flow theory. An $s$-$t$-cut is defined as a subset $S \subseteq V$ of nodes with $s \in S$ and $t \in V \setminus S$. We say that node $v$ is on the $s$-side of the cut if $v \in S$ and on the $t$-side of the cut if $v \in V \setminus S$. The capacity $\text{Cap}(S)$ of $S$ is defined as the sum of the capacities of the arcs going from the $s$-side to the $t$-side of the cut, that is, $\text{Cap}(S) := \sum_{e \in \delta^+(S)} u_e$, where $\delta^+(S)$ denotes the set of arcs $e = (v, w)$ with $v \in S$ and $w \in V \setminus S$. Throughout the paper, we use $F$ to denote the set of all $s$-$t$-cuts. An $s$-$t$-cut with minimum capacity is called a minimum cut. The minimum cut problem is to find a cut with minimum capacity. This problem
can be formulated as
\[
\begin{align*}
\min & \sum_{e \in \mathcal{E}} \rho_e u_e \\
\text{s.t.} & \quad \rho_e + \pi_v - \pi_w \geq 0 \quad \forall e = (v, w) \in \mathcal{E}, \\
& \quad \pi_s = 0, \quad \pi_t = 1, \\
& \quad \rho_e \geq 0 \quad \forall e \in \mathcal{E}.
\end{align*}
\]
(10)
Garg and Vazirani (1995) show that the extreme point solutions of this problem correspond exactly to \(s-t\)-cuts, in which the \(s\)-side is connected. In particular, if \(\pi, \rho\) is an extreme point solution, then \(\rho_e = \max\{0, \pi_v - \pi_w\}\), and \(\pi_v = 1\) or \(0\) for each \(v \in V\) \(\setminus \{s, t\}\). In this case, the binary variable \(\rho_e\) indicates whether or not arc \(e\) crosses the cut from \(S\) to its complement. The binary variable \(\mu_e\) indicates whether or not arc \(e\) fails. Hence, the corresponding \(s-t\)-cut is defined by \(S := \{v \in V \mid \pi_v = 0\}\).

Problem (10) is the dual of Problem (1). Because strong duality holds between these two problems, the value of a maximum flow equals the capacity of a minimum cut. Likewise, the robust maximum flow problem is formulated as Problem (5), which has the following corresponding dual problem:
\[
\begin{align*}
\min \sum_{e \in \mathcal{E}} \rho_e u_e \\
\text{s.t.} & \quad \rho_e + \pi_v - \pi_w \geq 0 \quad \forall e = (v, w) \in \mathcal{E}, \\
& \quad \pi_v - \sum_{e \in \delta^+(v)} \mu_e \geq 0 \quad \forall v \in V \setminus \{s\}, \\
& \quad \pi_s = 0, \quad \pi_t = 1.
\end{align*}
\]
Because Problem (5) is feasible and bounded, by strong duality the optimal objective function values of Problems (5) and (11) coincide.

In general, Problem (5), as well as Problem (11), may not have an integer-optimal solution. Consider a network of three nodes \(s, v, \) and \(t\). The arc set contains three parallel unit capacity arcs from \(s\) to \(v\), and likewise unit arcs from \(v\) to \(t\). For \(\Gamma_v = \Gamma_t = 2\), the optimal integral flow is 0. However, a fractional robust flow can send one unit of flow on each arc from \(s\) to \(v\) and \(1/2\) on each arc from \(v\) to \(t\). This is feasible as in any scenario at least one unit of flow reaches node \(v\). Consequently, the maximum robust flow value is \(1/3\). This is also the optimal fractional value for Problem (11) as it is the dual to Problem (5).

Still, we can characterize \(s-t\) cuts by means of Problem (11). If we restrict the feasible region of Problem (11) to binary solutions, any feasible solution corresponds to an \(s-t\)-cut defined by \(S := \{v \in V \mid \pi_v = 0\}\). Notice that in a binary optimal solution, we must have \(\rho_e = 0\) whenever \(\mu_e = 1\). This means that whenever arc \(e\) fails, it does not contribute to the objective function value. Under this condition, the objective function value of Problem (11) is equal to the sum of the capacities of the available arcs going from \(S\) to \(V \setminus S\). Hence, the following procedure gives the objective function value for a given cut \(S\): for each node \(v\) on the \(t\)-side of the cut \(S\) remove the—according to capacity—first \(\Gamma_v\) arcs crossing the cut into \(v\). Then, sum over all nodes \(v\) on the \(t\)-side of the cut \(S\) the capacities of all remaining arcs across \(S\) into \(v\). This intuition motivates to define the robust capacity of a cut \(S\) as
\[
\text{RCap}(S) := \sum_{e \in \mathcal{E}} u_e - \sum_{v \in V} z_v(S),
\]
where
\[
\begin{align*}
\text{z}_v(S) := & \max \sum_{e \in \delta^-(v) \cap \delta^+(S)} u_e \mu_e \\
\text{s.t.} & \quad \sum_{e \in \delta^-(v) \cap \delta^+(S)} \mu_e \leq \Gamma_v, \\
& \quad 0 \leq \mu_e \leq 1 \quad \forall e \in \delta^-(v) \cap \delta^+(S).
\end{align*}
\]
We have thus established the following result, which shows that there is a one-to-one correspondence between binary solutions of Problem (11) and \(s-t\)-cuts.

**Lemma 1.** Any binary solution of Problem (11) corresponds to an \(s-t\)-cut, whose objective function value is the same as the robust capacity, and vice versa.

A cut with minimum robust capacity is called a robust minimum cut. We define the robust minimum cut problem as follows.

**Robust Minimum Cut Problem**

*Input:* A network \(G = (V, E)\) with a source \(s \in V\), a sink \(t \in V\), arc capacities \(u_e \in \mathbb{R}_+\) for \(e \in E\), and a parameter \(\Gamma_v\) for each node \(v \in V \setminus \{s\}\).

*Task:* Find a robust minimum cut and its robust capacity.

Because the linear relaxation of the robust minimum cut problem is the dual problem to the robust maximum flow problem, its optimal value gives an upper bound on the value of any robust flow, i.e., \(\text{RVal}(x) \leq \text{RCap}(S)\) holds for each robust flow \(x\) and any \(s-t\)-cut \(S\). However, equality does not hold in general. To see this, consider as above an example of three nodes \(s, v, \) and \(t\). The arc set contains two parallel unit arcs from \(s\) to \(v\), and two from \(v\) to \(t\). Let \(\Gamma_v = \Gamma_t = 1\). There are two \(s-t\)-cuts with robust capacity equal to 1, whereas the optimal robust flow value is 1/2.

We next show that it is strongly NP-hard to compute a cut with minimum robust capacity. To prove this, we first show that the robust minimum cut problem is equivalent to the nodewise limited network interdiction problem:

**Node-Wise Limited Network Interdiction Problem**

*Input:* A network \(G = (V, E)\) with a source \(s \in V\), a sink \(t \in V\), arc capacities \(u_e \in \mathbb{R}_+\) for \(e \in E\), and a parameter \(\Gamma_v\) for each node \(v \in V \setminus \{s\}\).

*Task:* Minimize the maximum flow value by eliminating \(\Gamma_v\) arcs across each node \(v \in V \setminus \{s\}\).

**Lemma 2.** The robust minimum cut problem is equivalent to the nodewise limited network interdiction problem.
The robust minimum cut problem looks for a cut \( S \) for which the following value \( z \) is minimum: remove from each node \( v \) in the \( t \)-side of \( S \) the \( \Gamma_v \) incoming arcs with maximum capacities that cross the cut, and then \( z \) is the sum of the capacities of the remaining arcs across the cut. By the MaxFlow-MinCut Theorem, any flow through the network after this removal has a value less than or equal to \( z \). We can augment the set of removed arcs to a solution of the nodewise limited interdiction problem. Therefore, the optimal value of the interdiction problem is less than or equal to that of the minimum robust cut problem.

We proceed to show that the optimal value of the interdiction problem is greater than or equal to that of the minimum robust cut problem. Consider a set of optimal arcs to remove for the interdiction problem. The optimal value \( z' \) of the interdiction problem is that of a maximum flow through the remaining network. By the MaxFlow-MinCut Theorem, it equals the capacity of some cut \( S' \) in the remaining network. Therefore, the cut \( S' \) has a robust capacity of at most \( z' \) in the original network, which establishes the proof.

**Lemma 3.** The nodewise limited network interdiction problem is strongly NP-hard.

**Proof.** We use a reduction from the (nodewise limited) matching interdiction problem on bipartite graphs.

**Theorem 2.** The robust minimum cut problem is strongly NP-hard.
the sum of all failures, it is equivalent to having a local bound $\Gamma = \Gamma$ for all possible failures at $v$. Therefore, by a result from Bertsimas and Goyal (2010) the model of the previous section gives a constant factor approximation to the adjustable model, in case the center of the bounding box of the scenario set is a scenario itself. Note that this condition is not fulfilled unless $\Gamma \geq n/2$, with $n$ the total number of arcs.

The uncertainty sets we define here are similar in spirit to those used in Bertsimas and Sim (2003, 2004), because they use a parameter that limits the number of input values that can deviate. In our case, the right-hand side of the bound over the number of deviating matrix entries in one row, whereas they use a parameter that limits the number of input values within the interval $[0, u_e]$. In other words, at most $\Gamma$ arcs can fail. Still, there are a number of differences that inhibit the application of the results or methodology of Bertsimas and Sim (2003, 2004) here. Most importantly, in Bertsimas and Sim (2003, 2004), the parameter $\Gamma$ restricts the number of deviating matrix entries in one row, whereas here the restriction is to the number of rows that are subject to change.

Given $\mu \in \Lambda$, we denote by

$$E(\mu) := \{ e \in E \mid \mu_e = 0 \} = E \backslash \{ e \in E \mid \mu_e = 1 \}$$

the set of arcs after removing the arcs in the scenario $\mu$ and by $G(\mu) = (V, E(\mu))$ a network with arc set $E(\mu)$. Moreover, we let $N(\mu)$ as above be the node-arc adjacency matrix of $G(\mu)$ without the row of the sink. The artificial arcs $(t, s)$ never fails.

With respect to a given flow $x$ and scenario $\mu$, we denote by $G(x, \mu)$ a network with arc set $E(\mu)$ and arc capacities $x$. Moreover, we let $M(x, \mu)$ be the maximum flow value in $G(x, \mu)$, given by the following problem:

$$M(x, \mu) := \max \; y(s, t)$$

s.t.

$$\begin{align*}
  Ny &= 0, \\
  0 &\leq y_e \leq (1 - \mu_e)x_e & e \in E,
\end{align*}
$$

where $N$ is as before the matrix corresponding to the original network. In other words, $M(x, \mu)$ gives the maximum amount of flow that can be pushed through the network with respect to the flow $x$ if scenario $\mu$ occurs.

We define the adaptive value of $x$ as the amount of flow that can be pushed to the sink in the worst case, that is,

$$\text{AVal}(x) := \min_{\mu \in \Lambda} M(x, \mu).$$

We call an $s$-$t$-flow $x$ an adaptive maximum flow if it has the maximum adaptive value among all $s$-$t$-flows. The formal definition of the problem reads:

**Adaptive Maximum Flow Problem**

Input: A network $G = (V, E)$ with a source $s \in V$, a sink $t \in V$, arc capacities $u_e \in \mathbb{R}_+$ for $e \in E$ such that at most $\Gamma$ arcs can fail.

Task: Find an adaptive maximum flow and its adaptive value.

This problem can be formulated as follows:

$$\max_{x \in X} \min_{\mu \in \Delta} \max_{y \in \chi} y(s, t)$$

s.t.

$$\begin{align*}
  Ny &= 0, \\
  0 &\leq y_e \leq (1 - \mu_e)x_e & e \in E.
\end{align*}$$

Here and throughout the rest of the paper, we use $\chi$ to denote the set of all $s$-$t$-flows in the original network.

Before proceeding with the analysis, let us illustrate the concept of robust and adaptive maximum flows with a simple example. Consider the network depicted in Figure 2(a). The original network has five nodes and eight arcs. We extend the network by introducing an artificial arc $(t, s)$ with infinite capacity. Figure 2(d) denotes a maximum flow for the nominal problem, in which the flow on arc $(t, s)$ is the flow value. It is obvious that the zero flow is the only robust flow even for $\Gamma = 1$. This is always the case if there are no parallel arcs out of the source because for any node $v$ with $(s, v) \in E$, no flow can be sent out of $v$ when arc $(s, v)$ disappears. To avoid this trivial situation, we may assume that either there are parallel arcs out of $s$ or the arcs out of the source are not subject to failure. In this example, we consider the latter assumption. The robust maximum flows are shown in Figures 2(b) and 2(c) for $\Gamma = 1$ and $\Gamma = 2$, respectively, and the flow on arc $(t, s)$ shows the value of the corresponding robust flow. Figures 2(e) and 2(f) show optimal flows for the adaptive maximum flow problem for $\Gamma = 1$ and $\Gamma = 2$, in which the optimal values are 40 and 20, respectively.

The problem of computing $\text{AVal}(x)$ for a given $s$-$t$-flow $x$ reduces to the problem of finding the $\Gamma$ most vital arcs in a network with arc capacities $x_e, e \in E$, i.e., finding the $\Gamma$ arcs whose removal from the network minimizes the maximum amount of flow that can be sent to the sink through the network. This problem is known as the network interdiction problem (see, e.g., Wood 1993), which asks to reduce the maximum flow value in a given network as much as possible by removing $\Gamma$ arcs. The adaptive maximum flow problem seeks a flow whose adaptive value after removing $\Gamma$ arcs in the worst case is maximized. In other words, it seeks an arc vector that is a feasible flow and has the best network interdiction value among all feasible flows.

The example in Figure 3 illustrates the differences between the network interdiction problem and the adaptive maximum flow problem. The numbers on the arcs indicate the capacities. Suppose that $\Gamma = 1$. Obviously, the maximum flow value for the nominal problem is 3. For the network interdiction problem, if we remove one of the arcs $e_1$, $e_2$, and $e_3$, the maximum flow value in the remaining network is 2. If we remove one of the arcs $e_4$ or $e_5$, the maximum flow value is still 3. Therefore, each of the arcs $e_1$, $e_2$, or $e_3$ is the most vital arc and the optimal value for...
Figure 2. Illustration of the concept of robust and adaptive maximum flows.

(a) Arc capacities.

(b) A robust maximum flow with value 30 for $\Gamma = 1$.

(c) A robust maximum flow with value 10 for $\Gamma = 2$.

(d) A maximum flow for the nominal maximum flow problem.

(e) An adaptive maximum flow with value 40 for $\Gamma = 1$.

(f) An adaptive maximum flow with value 20 for $\Gamma = 2$.

Note. It is assumed that the arcs out of $s$ are not allowed to fail.

It is known from network flow theory that if all arc capacities are integer, the maximum flow problem always has a maximum integer flow. The example in Figure 3 also shows that this is not the case in general for adaptive maximum flows.

In general, there might be no $s$-$t$-flow, which simultaneously maximizes the nominal and adaptive values (see the example in Figure 4). However, we can prove the existence of an adaptive maximum flow, which is also maximum for the nominal problem if either $\Gamma = 1$ or $G$ is a series-parallel graph.

**Lemma 4.** If either $\Gamma = 1$ or $G$ is a series-parallel graph, there always exists a flow that simultaneously maximizes both the nominal and the adaptive values.

**Proof.** Let $x$ be an adaptive maximum flow, which is not optimal for the nominal problem. Hence, there are augmenting paths in the residual graph with respect to flow $x$ and we can improve the flow value by sending flow along these paths. In general, sending flow along an augmenting...
There might be no $s$-$t$ flow, which is simultaneously optimal for both the nominal problem and the adaptive maximum flow problem.

Figure 4. There might be no $s$-$t$ flow, which is simultaneously optimal for both the nominal problem and the adaptive maximum flow problem.

Notes. The numbers on the arcs indicate the capacities. There exists a unique maximum flow that sends four units of flow from source $s$ to sink $t$. Suppose that $\Gamma = 2$. The adaptive maximum flow value is 1, and there is no robust maximum flow that is also maximum for the nominal problem.

path may decrease the adaptive value. We show that this is not the case under either of the hypotheses of the lemma.

Case: $\Gamma = 1$. We first assume that $\Gamma = 1$ and show that the adaptive value is not decreased by sending flow along an augmenting path. Let $P$ be an augmenting path from $s$ to $t$. For every $s$-$t$-cut $S$, the path $P$ contains one more arc leaving $S$ than entering $S$, i.e., to get from $s$ to $t$ the path has to leave the cut once more than it enters the cut. Therefore, if an augmenting path decreases the flow on $k$ arcs of a cut by a value of $\kappa$, then it increases the flow on the $k + 1$ arcs of the same cut by a value of $\kappa$.

Now let $x'$ be the flow after augmenting $x$ maximally along the path $P$. Further, let $S$ be a cut and $\mu$ be some scenario. By hypothesis in scenario $\mu$, at most one arc fails. Assume that the augmenting path $P$ uses $e$, the failing arc of scenario $\mu$, as a forward arc, and $e'$ is an arc leaving $S$. Among the other arcs crossing the cut $S$ equally, many are used as forward and as backward arcs by the augmenting path $P$. Therefore, after the augmentation along $P$, the capacity of $S$ in $G(x', \mu)$ (i.e., $S'(e)$) is still not less than that of cut $S$ in $G(x, \mu)$. This is, of course, still true, in case $e'$ is used as a backward arc or is not in $P$.

Thus, for no cut the minimal capacity over all scenarios decreases from $G(x)$ to $G(x')$—in fact, it decreases for no scenario. Therefore, in every scenario $\mu$ there is a flow in $G(x', \mu)$ with at least the maximum flow value of $G(x, \mu)$.

Case: Series-Parallel. We now assume that $G$ is a series-parallel graph. We recall that series-parallel graphs are graphs with two distinguished nodes formed recursively by two simple composition operations, series composition and parallel composition (see, e.g., Eppstein 1992). It is known in series-parallel graphs that a flow is maximum if and only if it is a blocking flow. We notice that a flow is a blocking flow if every path from $s$ to $t$ contains a saturated arc (an arc whose flow equals its capacity). In general, a blocking flow is not necessarily a maximum flow, because there might exist a path from $s$ to $t$ in the residual network. However, this is not the case in series-parallel graphs (see, e.g., Williamson 2004).

Therefore, in a nonmaximum flow there are augmenting paths with respect to $x$, containing only forward arcs. We can improve the value of $x$ by sending flow along these paths until a blocking flow is obtained. Obviously, sending flow along an augmenting path containing only forward arcs does not reduce the adaptive value. Therefore, the resulting flow will be a maximum flow, whose adaptive value is not less than that of $x$. □

We next proceed to give an alternative formulation of Problem (15). We first show that this problem can be expressed as an LO problem with (potentially exponentially) many constraints and variables. For a given $s$-$t$-flow $x$, the adaptive value $\text{AVal}(x)$ can be expressed as

$$\text{AVal}(x) = \max z$$

s.t. $z \leq M(x, \mu)$ $\forall \mu \in \Lambda$. (16)

For a scenario $\mu \in \Lambda$, we have $z \leq M(x, \mu)$ if and only if there is an $s$-$t$-flow $y^{\mu}$ in the network $G(x, \mu)$ whose value is at least $z$. Therefore, the above problem can be rewritten as follows:

$$\text{AVal}(x) = \max z$$

s.t. $z - y^{\mu}_{(r,s)} \leq 0$ $\forall \mu \in \Lambda$, $N(\mu)y^{\mu} = 0$ $\forall v \in V \setminus \{s, t\}, \mu \in \Lambda$, $0 \leq y^{\mu} \leq x$ $\forall e \in E, \mu \in \Lambda$. (17)

In this problem, $y^{\mu}$ is an $s$-$t$-flow in the network $G(x, \mu)$ due to the second and the third sets of constraints. Moreover, the first set of constraints implies that $z \leq \text{Val}(y^{\mu})$ for all $\mu \in \Lambda$, because the objective function is to maximize $z$, and hence the adaptive value of $x$. Hence, Problem (15) can be simplified as

$$\max z$$

s.t. $N x = 0$, $0 \leq x \leq u$, (18)

constraints of Problem (17).

The number of constraints and variables in this problem depends linearly on the number of scenarios. If $\Gamma$ is constant (i.e., it is not part of the input), the above problem has a polynomial number of variables and constraints and can be solved in polynomial time. However, if $\Gamma$ is part of the input, then the problem can have exponentially many variables and constraints. The above observations show that the adaptive maximum flow problem is solvable in polynomial time when $\Gamma$ is constant. In the next section, we show that the adaptive maximum flow problem is NP-hard for $\Gamma$ part of the input.
3.2. Hardness Results

Using the equivalence of optimization and separation (see Grötschel et al. 1988), Problem (15) can be solved in polynomial time if and only if the following separation problem can be solved in polynomial time: given an $s$-$t$-flow $x$ and a value $z$, determine whether or not $AVal(x) \geq z$. In other words, the separation problem asks whether or not the inequality $M(x, \mu) \geq z$ holds for all $\mu \in \Lambda$.

The separation problem asks to determine the adjustable robust value of a given flow. This is an instance of the network interdiction problem, if the flow on each arc is treated as its capacity. Therefore, the separation problem of the adaptive maximum flow problem is the network interdiction problem restricted to the case, where the arc capacities fulfill strict flow conservation. In other words, only those instances occur, where the sum of arc capacities into a node equals the sum of arc capacities out of that node, for all nodes except for the terminals.

Wood (1993) shows that the network interdiction problem is strongly NP-hard by a reduction from the Maximum clique problem. Prima facie, this does not imply the hardness of the separation problem, because it is a special case of the network interdiction problem. However, we can modify Wood’s reduction to show that the interdiction problem is still NP-hard if the arc capacities satisfy flow conservation. This implies that computing the adaptive value of a flow is NP-hard. In fact, we can show that any inapproximability result for network interdiction immediately carries over to adjustable robust flows with the same factor. Complementary to this, we note that any approximation algorithm for the separation problem—e.g., an approximation algorithm for network interdiction—yields an approximation of the adaptive flow.

**Lemma 5.** For every $\alpha \geq 1$, such that it is NP-hard to $\alpha$–approximate the network interdiction problem, it is NP-hard to $\alpha$–approximate the adaptive value of a given flow.

**Proof.** We show that every instance of network interdiction constructed in the hardness proof for network interdiction by Wood (1993) is either solvable in polynomial time, or by reducing it to finding the value of an adaptive flow. For every instance $(G, u, \Gamma)$ of the network interdiction problem, given by an $s$-$t$–digraph $G$ with arc capacities $u$ and an integer bound $\Gamma$ on the number of failing arcs, we construct an adaptive $s$-$t$-flow $x$ on a digraph $G'$ and an integer bound $\Gamma'$ on the number of failing arcs $\Gamma' = \Gamma + 2$. The graph $G'$ will contain the graph $G$, and we set $x = u$ on $G$. Now, every node in $V(G') \setminus \{s, t\}$ either fulfills flow conservation or has an excess of inflow over outflow, or vice versa. We introduce two new nodes $p$ and $q$ and connect each node $v$ with a surplus of inflow to the node $p$ and set $x_{(v, p)}$ equal to the surplus. Likewise, we connect node $q$ to each node $w$ with a surplus in outflow, and set $x_{(q, w)}$ equal to that surplus. To complete the construction of $G'$, we add the arcs $(s, p)$, $(s, q)$, $(p, t)$, and $(q, t)$. The construction of graph $G'$ is depicted in Figure 5.

**Figure 5.** Construction of the graph $G'$ from $G$.

We choose the values $x_{(s, p)}$ and $x_{(q, t)}$ sufficiently large, (e.g., $4\alpha$ times the maximum flow value in $G$) and set $x_{(p, t)} = x_{(s, p)} + \sum_{v \in V(G)} x_{(v, p)}$, respectively, $x_{(s, q)} = x_{(q, t)} + \sum_{v \in V(G)} x_{(q, w)}$, so that $(p, t)$ and $(s, q)$ are part of any optimal network interdiction set. By construction, $x$ is a flow.

Now, we split the arcs $(s, p)$ and $(q, t)$ into $2x_{(s, p)}$-many, respectively, $2x_{(q, t)}$-many arcs of capacity 0.5. Thereby, both $(s, p)$ and $(q, t)$ become replaced by an infallible set of arcs, because no optimal network interdiction solution will use any of these arcs. We show below why this can be assumed. The network in our construction has half integral arc capacities. Finally, doubling all arc capacities gives an equivalent problem with integral capacities. Note that the construction in the proof of Wood (1993) uses only arc capacities equal to 1 or 2. Thereby, the maximum flow value is polynomially bounded by the number of arcs in $G$. Accordingly, the number of arcs in $G'$ after the above splitting of arcs $(s, p)$ and $(q, t)$ is still polynomial in the number of arcs in $G$. Hence, we give a strongly polynomial reduction in total.

As $x_{(s, q)}$ and $x_{(p, t)}$ are sufficiently large, any $\alpha$–approximation to the adaptive value of $x$ with $\Gamma'$ failing arcs, must fail the arcs $(s, q)$ and $(p, t)$. Thus, the optimal adaptive value of $(G', x, \Gamma')$ is equal to the optimal network interdiction value of $(G, u, \Gamma)$, and every $\alpha$–approximation to the adaptive value problem is an $\alpha$–approximation to the network interdiction problem.

**Infallible arcs:** We now show that one can assume that no arc of unit capacity is part of any optimal solution to the network interdiction problem on the constructed network. Firstly, consider any $s$-$t$-cut $S$ that is minimal after an optimal network interdiction. The arcs crossing the cut $S$ in the uninterdicted network are exactly those that cross $S$ after the interdiction plus those of the interdiction set. In fact, the interdiction set contains exactly the $\Gamma$ largest arcs across $S$. Else, exchanging a smaller arc crossing $S$, or an arc not crossing $S$ for one of the $\Gamma$ largest arcs across $S$ would decrease the value of the minimum cut $S$ after the interdiction, contradicting optimality.

Secondly, as the network constructed here has capacities equal to the values of a flow, we know that every cut is a
minimum cut, i.e., all cuts have the same value. Combining the two observations, we see that the network interdiction set will contain no unit capacity arc, unless there is a cut containing \((s, q)\) and \((p, t)\) (which must be part of any optimal interdiction) with less than \(\Gamma + 2\) arcs of higher capacity. The latter would require the original network \(G\) to have a cut of fewer than \(\Gamma\) arcs, i.e., its interdiction value is 0. Hence, we check every network interdiction instance for having value 0 before we apply the transformation. This can be done in polynomial time: Set all arc capacities in \(G\) to 1, and check whether then the maximum flow value is less than or equal to \(\Gamma\). Because these network interdiction instances are polynomial-time solvable, we can exclude them from the reduction. Therefore, we can assume that the arcs replacing \((s, p)\) and \((q, t)\) are inadmissible. □

The network interdiction problem is NP-hard in the strong sense (see Wood 1993). Thus, as an immediate consequence of Lemma 5, we get the following result.

**Corollary 1.** The adaptive maximum flow problem is strongly NP-hard.

Although the network interdiction problem is NP-hard in general, Wollmer (1964) presents a polynomial-time algorithm to solve the problem on \(s\)-\(t\)-planar graphs. This implies that we can compute the adaptive value of \(x\) in polynomial time on \(s\)-\(t\)-planar graphs. Using the equivalence of optimization and separation (see Grötschel et al. 1988), this proves that the adaptive maximum flow problem is solvable in polynomial time on \(s\)-\(t\)-planar graphs. We are not aware of any bounds on the approximability of the network interdiction problem, and neither of any approximation algorithms for it. Any such bound or approximation algorithm would immediately carry over to the adaptive maximum flow problem.

The adaptive maximum flow is not integral in general and cannot be formulated as an LO problem of polynomial size. In fact, unless \(P = NP\) there is no LP formulation for the adjustable maximum flow problem because it is NP-hard. However, we will define a problem, the adaptive minimum cut, that is strongly dual to the adaptive maximum flow in the sense that their optimal objective values coincide (see Theorem 4). If a problem \(\Pi\) is in \(NP\) and there is a strongly dual problem to it in \(NP\), then \(\Pi\) is in \(coNP\). Problems that are in \(NP\) and in \(coNP\) are usually polynomially solvable. However, the adaptive flow problem is NP-hard. Indeed, the adaptive maximum flow problem can be formulated as an \(L_\infty\) problem with a separation oracle (see Problem 18) that solves an NP-hard problem. In other words, the adaptive maximum flow problem is in \(p^{NP}\), which is \(\Sigma^p_2\) in the polynomial hierarchy. Although there is no proof for being \(\Sigma^p_2\)-complete it might well be outside \(NP\). This can explain why the adaptive flow can have a strong dual while being NP-hard.

### 3.3. Quadratic and Integer Optimization Problems

As mentioned before, the problem of computing the adaptive value of a flow \(x\) reduces to the network interdiction problem by considering the flow \(x_e\) as the capacity of arc \(e\). Wood (1993) formulates an integer LO problem for the network interdiction problem. Altner et al. (2010) prove that the integrality gap of the LP relaxation of Wood’s integer optimization problem cannot be bounded by a constant factor, even when strengthened by certain two classes of polynomially separable valid inequalities.

In what follows, we give an alternative binary optimization formulation for computing the adaptive value of a flow \(x\) as well as for the network interdiction problem. We first show how Problem (14) can be expressed as a quadratic optimization problem and further derive the integer optimization problem.

**Lemma 6.** Let \(x\) be an \(s\)-\(t\)-flow. The adaptive value of \(x\) can be computed by the following quadratic optimization problem:

\[
\text{AVal}(x) = \min \sum_{e \in E} \rho_e (1 - \mu_e) x_e
\]

s.t. \(\rho_e + \pi_e - \pi_w \geq 0 \quad \forall e = (v, w) \in E,\)

\[
\sum_{e \in E} \mu_e \leq \Gamma,
\]

\[
0 \leq \mu_e \leq 1 \quad \forall e \in E,
\]

\[
\pi_s = 0, \quad \pi_t = 1,
\]

\[
\rho_e \geq 0 \quad \forall e \in E,
\]

\[
\pi_v \geq 0 \quad \forall v \in V.
\]

**Proof.** Recall that \(\text{AVal}(x) = \min_{\mu \in \mathcal{A}} M(x, \mu)\) where \(M(x, \mu)\) is given by Problem (13). We write the dual problem of Problem (13) and obtain

\[
M(x, \mu) = \min \sum_{e \in E} (1 - \mu_e) x_e \rho_e
\]

s.t. \(\rho_e + \pi_e - \pi_w \geq 0 \quad \forall e = (v, w) \in E,\)

\[
\pi_s = 0, \quad \pi_t = 1,
\]

\[
\rho_e \geq 0 \quad \forall e \in E.
\]

The function \(M(x, \mu)\) is concave in \(\mu\) because it is the pointwise minimum of linear functions of \(\mu\) (see, e.g., Boyd and Vandenberghe 2004). A concave function over a compact domain attains its minimum at an extreme point of the feasible region. Hence, we can relax the binary variables \(\mu\) to be continuous as the extreme points of the polyhedron \(\{\mu \in [0, 1]^{|E|} : \sum_{e \in E} \mu_e \leq \Gamma\}\) are binary. This completes the proof of the lemma. □

The objective function of Problem (19) is quadratic and concave. In general, it is NP-hard to optimize such a function over a polytope. However, there are approximation algorithms for solving a concave quadratic optimization problem (see, e.g., Pardalos and Rosen 1987 and Ye 1992), and we can use these existing techniques to approximate...
the optimal solutions of Problem (19). For practical computations, we reformulate Problem (19) as an integer optimization problem. Let us consider the following quadratic optimization problem:

\[
\begin{align*}
\min & \quad \frac{1}{2} x^T Q x + c^T x \\
\text{s.t.} & \quad Ax \geq b, \\
& \quad x \geq 0,
\end{align*}
\]

(21)

where \( A \) is an \( m \times n \) matrix, \( c \) and \( x \) are \( n \)-vectors, \( b \) is an \( m \)-vector, and \( Q \) is an \( n \times n \) matrix. All vectors are supposed column vectors. Using the Karush–Kuhn–Tucker conditions, Problem (21) can be written as follows:

\[
\begin{align*}
\min & \quad \frac{1}{2} (c^T x + \lambda^T b) \\
\text{s.t.} & \quad Q x + A^T \lambda - y = -c^T, \\
& \quad Ay + w = b, \\
& \quad y^T x = \lambda^T w = 0, \\
& \quad x, \lambda, y, w \geq 0,
\end{align*}
\]

where \( \lambda \) and \( w \) are \( m \)-vectors and \( y \) is a \( n \)-vector. Here the objective function and the first two sets of constraints are linear. The third set of constraints enforces the complementary slackness conditions. By applying this approach to Problem (19), we obtain an LO model except for the complementary slackness conditions. However, the constraints enforcing complementary slackness can be turned into linear constraints by introducing binary variables. Further, the complementary slackness conditions can be modeled as Special Ordered Set (SOS) constraints. These kinds of constraints are efficiently implemented by existing mixed-integer optimization solvers like Gurobi or CPLEX. Another possibility is to use the simplex algorithm on the remaining LO model and treat the complementary slackness conditions implicitly with a restricted basic entry rule (see, e.g., Jensen and Bard 2003 for more details).

### 3.4. Adaptive Minimum Cut Problem

Here we define the adaptive minimum cut \( \text{cut} \) and investigate its relationship to the adaptive maximum flow problem. We start with the definition of the adaptive capacity of an \( s-t \)-cut. Given an \( s-t \)-cut \( S \) and an \( s-t \)-flow \( x \), we define \( R(x, S) \) as the amount of flow that can pass through the cut \( S \) in the worst case, that is,

\[
R(x, S) := \min_{\mu \in \Lambda} \sum_{e \in \delta^+(S)} (1 - \mu_e) x_e.
\]

(22)

Based on this definition, we can provide an alternative formula for computing the adaptive value of a flow as described in the following lemma.

**Lemma 7.** Suppose that \( x \) is an \( s-t \)-flow. Then, we have

\[
\text{AVal}(x) = \min_{S \in \mathcal{F}} R(x, S).
\]

**Proof.** Recall that \( \text{AVal}(x) = \min_{\mu \in \Lambda} M(x, \mu) \), where \( M(x, \mu) \) is the maximum flow value in the network \( G \) with respect to arc capacities \((1 - \mu_e)x_e\). Using the MaxFlow-MinCut Theorem for standard network flows, we have

\[
M(x, \mu) = \min_{S \in \mathcal{F}} \sum_{e \in \delta^+(S)} (1 - \mu_e) x_e,
\]

and consequently

\[
\text{AVal}(x) = \min_{\mu \in \Lambda} M(x, \mu) = \min_{\mu \in \Lambda} \min_{S \in \mathcal{F}} \sum_{e \in \delta^+(S)} (1 - \mu_e) x_e
\]

\[
= \min_{S \in \mathcal{F}} \min_{\mu \in \Lambda} \sum_{e \in \delta^+(S)} (1 - \mu_e) x_e
\]

\[
= \min_{S \in \mathcal{F}} R(x, S). \quad \Box
\]

We define the adaptive capacity \( \text{ACap}(S) \) of an \( s-t \)-cut \( S \) as

\[
\text{ACap}(S) := \max_{x \in \mathcal{F}} R(x, S).
\]

Although it is NP-hard to find the adaptive value of a fixed flow, the following lemma shows that the adaptive capacity of a given cut can be computed in polynomial time.

**Lemma 8.** Let \( S \) be an \( s-t \)-cut. The adaptive capacity \( \text{ACap}(S) \) of \( S \) can be obtained by the following LO model:

\[
\text{ACap}(S) = \max \sum_{e \in \delta^+(S)} (x_e - \beta_e) - \Gamma \delta
\]

\[
\text{s.t.} \quad \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0, \\
& \quad \forall v \in V \setminus \{s, t\}, \\
& \quad 0 \leq x_e \leq u_e \quad \forall e \in E, \\
& \quad \delta + \beta_e \geq x_e \quad \forall e \in \delta^+(S), \\
& \quad \beta_e \geq 0 \quad \forall e \in \delta^+(S), \\
& \quad \delta \geq 0.
\]

(23)

**Proof.** Clearly, the binary variables \( \mu_e \) for all \( e \in \delta^+(S) \) in Problem (22) can be relaxed to be continuous. For a given \( s-t \)-flow \( x \), we consider the dual problem of Problem (22) and obtain:

\[
R(x, S) = \max \sum_{e \in \delta^+(S)} (x_e - \beta_e) - \Gamma \delta
\]

\[
\text{s.t.} \quad \delta + \beta_e \geq x_e \quad \forall e \in \delta^+(S), \\
& \quad \beta_e \geq 0 \quad \forall e \in \delta^+(S), \\
& \quad \delta \geq 0.
\]

(24)

The result now follows from the fact that \( \text{ACap}(S) = \max_{x \in \mathcal{F}} R(x, S) \). \( \Box \)
An $s$-$t$-cut $S$ whose adaptive capacity is minimum among all $s$-$t$-cuts is called an adaptive minimum cut. We can now give a formal description of the adaptive minimum cut problem.

**Adaptive Minimum Cut Problem**

Input: A network $G = (V, E)$ with a source $s \in V$, a sink $t \in V$, arc capacities $u_e \in R_+$, $e \in E$ such that at most $\Gamma$ arcs can fail.

Task: Find an adaptive minimum cut and its adaptive capacity.

**Theorem 3.** The adaptive minimum cut problem can be formulated as follows:

$$\min \sum_{e \in E} (1 - \rho_e) \beta_e u_e + \sum_{e \in E} \lambda_e u_e$$

s.t. $\beta_e + \lambda_e + \alpha_v - \alpha_w - \rho_e \geq 0 \quad \forall e = (v, w) \in E,$

$$\rho_e + \pi_v - \pi_w \geq 0 \quad \forall e = (v, w) \in E,$$

$$\sum_{e \in E} \beta_e \rho_e \leq \Gamma,$$

$$0 \leq \eta_e \leq 1 \quad \forall e \in E,$$

$$\beta_e + \lambda_e \geq 0 \quad \forall e \in E,$$

$$\rho_e \in \{0, 1\} \quad \forall e \in E,$$

$$\pi_v \in \{0, 1\} \quad \forall v \in V,$$

$$\delta \geq 0,$$

$$\pi_s = 0, \quad \pi_t = 1,$$

$$\alpha_s = \alpha_t = 0.$$  \hfill (25)

**Proof.** For a given $s$-$t$-cut $S$, we define $\pi = (\pi_e)_{e \in E}$ with $\pi_e := 0$ if $v$ is on the $s$-side of the cut and $\pi_e := 1$ if $v$ is on the $t$-side of the cut. Moreover, for each arc $e$, we let $\rho_e := 1$ if it crosses the cut, and $\rho_e := 0$ otherwise. This implies that $\rho_e = \max\{0, \pi_w - \pi_v\}$ for each $e = (v, w) \in E$. We define

$$h(\pi) := \max \sum_{e \in E} \rho_e (x_e - \beta_e) - \Gamma \delta$$

s.t. $\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s, t\},$

$$x_e - \rho_e (\delta + \beta_e) \leq (1 - \rho_e) u_e \quad \forall e \in E,$$

$$0 \leq x_e \leq u_e \quad \forall e \in E,$$

$$\beta_e \geq 0 \quad \forall e \in E,$$

$$\delta \geq 0.$$  \hfill (26)

By Lemma 8 we have $\Lambda \text{Cap}(S) = h(\pi)$. Considering the dual of Problem (26), we obtain

$$h(\pi) = \min \sum_{e \in E} (1 - \rho_e) u_e \eta_e + \sum_{e \in E} u_e \lambda_e$$

s.t. $\lambda_e + \eta_e + \alpha_v - \alpha_w \geq \rho_e,$

$$\forall e = (v, w) \in E,$$

$$\sum_{e \in \delta^+(v)} \lambda_e - \sum_{e \in \delta^-(v)} \lambda_e = 0 \quad \forall v \in V \setminus \{s, t\},$$

$$\sum_{e \in E} \rho_e \eta_e \leq \Gamma,$$

$$0 \leq \eta_e \leq 1 \quad \forall e \in E,$$

$$\beta_e, \lambda_e \geq 0 \quad \forall e \in E,$$

$$\delta \geq 0,$$

$$\alpha_s = \alpha_t = 0.$$  \hfill (27)

The adaptive minimum cut problem is to minimize $h(\pi)$ over $\{\pi | \pi_e \in \{0, 1\}^V, \pi_s = 1, \pi_t = 0\}$. In the resulting problem, the variable $\rho_e$ appears in the objective function with a negative coefficient. Thus, the requirement $\rho_e = \max\{0, \pi_w - \pi_v\}$ can be relaxed to $\rho_e \geq 0$ and $\rho_e \geq \pi_w - \pi_v$ for each $e = (v, w) \in E$. This completes the proof of the theorem. \hfill \square

In the following, we show that the adaptive maximum flow problem and the adaptive minimum cut problem can be modeled as a two-person zero-sum game. We consider two players called player 1 and player 2, and assume that the set of pure strategies of player 1 is given by $X$, and the set of pure strategies of player 2 is given by $Y$. If player 1 chooses the pure strategy $x \in X$ and player 2 chooses the pure strategy $y \in Y$, then player 1 gains an amount $R(x, y) = R(x, s)$, as given by Problem (22). This function is called the payoff function of player 1. Clearly, player 1 likes to gain as much flow as possible. A bound lower on his payoff is given by

$$\max_{x \in X} \min_{y \in Y} R(x, y),$$

which is equal to the adaptive maximum value. The payoff function of player 2 is $-R(x, y)$, because the gain of player 1 is the loss of player 2. An upper bound on the loss of player 2 is given by

$$\min_{y \in Y} \max_{x \in X} R(x, y),$$

which is equal to the adaptive minimum capacity. Obviously, we have

$$\max_{x \in X} \min_{y \in Y} R(x, y) \leq \min_{y \in Y} \max_{x \in X} R(x, y).$$  \hfill (28)

We show that equality holds if we extend the set of pure strategies to the larger set of so-called mixed strategies.

A mixed strategy is an assignment of a probability to each pure strategy. In other words, in a mixed strategy a player selects randomly among a set of pure strategies. We only consider the case that the set of pure strategies with positive probability is finite and denote the set of mixed strategies for player 1 by $F(X)$ and the set of mixed strategies for player 2 by $F(Y)$. We need to extend the payoff function $R$ to the Cartesian product of the sets $F(X)$ and $F(Y)$. Let $\alpha \in F(X)$ and $\beta \in F(Y)$. Formally, for any $\alpha \in F(X)$, there exists some finite set $\{x^1, \ldots, x^l\} \subset X$ and a sequence $\alpha_i, 1 \leq i \leq l$, satisfying

$$\sum_{i=1}^l \alpha_i = 1 \quad \alpha_i > 0, \quad 1 \leq i \leq l.$$
Similarly, for $\beta \in F(\mathcal{I})$, there exists some finite set $\{S_1, \ldots, S_k\} \subseteq \mathcal{I}$ and a sequence $\beta_j$, $1 \leq i \leq k$, satisfying
\[
\sum_{j=1}^{k} \beta_j = 1 \quad \beta_j > 0, \quad 1 \leq j \leq k.
\]

We now define the expected payoff function $R_c: F(\mathcal{I}) \times F(\mathcal{I}) \rightarrow \mathbb{R}$ by
\[
R_c(\alpha, \beta) := \sum_{i=1}^{l} \sum_{j=1}^{k} \alpha_i \beta_j R(x^i, S_j).
\]

If player 1 chooses a pure-strategy $x$ and player 2 chooses a mixed-strategy $\beta$, we denote the expected payoff by
\[
R_c(x, \beta) := \sum_{j=1}^{k} \beta_j R(x, S_j).
\]

We next establish the existence of an equilibrium for the adaptive maximum flow player and the adaptive minimum cut player, in which player 1 chooses a pure strategy and player 2 chooses a mixed strategy.

**Theorem 4.** We have
\[
\max \min_{x \in X} R_c(x, \beta) = \min \max_{\beta \in F(\mathcal{I})} R_c(x, \beta).
\]

**Proof.** Because $\mathcal{I}$ is a finite set, it follows from Wald’s Minimax Theorem (see, e.g., Wald 1945 and Frenk et al. 2004) that
\[
\max_{\alpha \in F(\mathcal{X})} \min_{\beta \in F(\mathcal{I})} R_c(\alpha, \beta) = \min_{\beta \in F(\mathcal{I})} \max_{\alpha \in F(\mathcal{X})} R_c(\alpha, \beta).
\]

Therefore, that there are $s$-t-flows $\bar{x}^1, \ldots, \bar{x}^l$, $s$-t-cuts $\bar{S}_1, \ldots, \bar{S}_k$, and nonnegative numbers $\bar{\alpha}_1, \ldots, \bar{\alpha}_l$ and $\bar{\beta}_1, \ldots, \bar{\beta}_k$ with $\sum_{i=1}^{l} \bar{\alpha}_i = 1$ and $\sum_{j=1}^{k} \bar{\beta}_j = 1$ such that
\[
\max_{\alpha \in F(\mathcal{X})} \min_{\beta \in F(\mathcal{I})} R_c(\alpha, \beta) = \min_{\beta \in F(\mathcal{I})} \max_{\alpha \in F(\mathcal{X})} R_c(\alpha, \beta)
\]
\[
= \sum_{i=1}^{l} \sum_{j=1}^{k} \bar{\alpha}_i \bar{\beta}_j R(\bar{x}^i, \bar{S}_j).
\]

We define $x^* := \sum_{i=1}^{l} \bar{\alpha}_i \bar{x}^i$ and set $S^* := \arg \min_{x \in \mathcal{X}} R(x^*, S)$. We show that $AVal(x^*) = ACap(S^*)$. We can write
\[
\max_{x \in X} \min_{\beta \in F(\mathcal{I})} R_c(x, \beta) \geq \min_{\beta \in F(\mathcal{I})} \sum_{i=1}^{q} \beta_i R(x^i, S_j)
\]
\[
\geq \min_{\beta \in F(\mathcal{I})} \sum_{i=1}^{q} \sum_{j=1}^{k} \bar{\alpha}_i \beta_j R(\bar{x}^i, \bar{S}_j)
\]
\[
= \sum_{i=1}^{k} \sum_{j=1}^{k} \bar{\alpha}_i \beta_j R(\bar{x}^i, \bar{S}_j)
\]
\[
= \max_{\alpha \in F(\mathcal{X})} \min_{\beta \in F(\mathcal{I})} R_c(\alpha, \beta),
\]
where the second inequality follows from the fact that $R(x, S)$ is concave in $x$ since it is the pointwise minimum of linear functions of $x$ (see, e.g., Boyd and Vandenberghe 2004).

On the other hand, we have
\[
\max \min_{x \in X} R_c(x, \beta) \leq \min \max_{\beta \in F(\mathcal{I})} R_c(x, \beta)
\]
\[
\leq \min \max_{\beta \in F(\mathcal{I})} R_c(\alpha, \beta).
\]

The desired result follows from (31) and (32).

**4. Robust Maximum Flows in Path-Based Formulation**

So far we have focused on $s$-t-flows based on arc flows. One may also define $s$-t-flows on paths and cycles, instead of arcs, leading to the path-based formulation of flows. This alternative definition will help us to present a different notion of robust maximum flows. Let $P$ and $C$ denote the sets of all $s$-t-paths and all cycles in $G$, respectively. For $P \in P \cup C$, we write $e \in P$ to indicate that arc $e \in E$ lies on the path or cycle $P$. Given an $s$-t-flow $x$, it is a well-known result form network flow theory that $x$ has a flow decomposition $(x_P)_{P \in P \cup C}$, where $x_P \geq 0$ for each $P \in P \cup C$ and $x = \sum_{P \in P \cup C} x_P$ for each $e \in E$. Moreover, there always exists a flow decomposition for which the number of flow-carrying paths and cycles, i.e., paths and cycles $P$ with $x_P > 0$, is bounded by the number of arcs in the network. We refer to $x_P$ as the flow on path or cycle $P$. Clearly, the value of $x$ is the sum of the flows on the flow-carrying paths, i.e., $Val(x) = \sum_{P \in P \cup C} x_P$. Flows on cycles do not contribute to the value of $x$. Therefore, we can assume that $x_P = 0$ for all cycles $P \in C$. A flow decomposition with this property is called a path decomposition.

In the path-based formulation we associate a value to each path. A flow is feasible if the total flow on each arc does not exceed the capacity of the arc. It is clear that the flow conservation constraint is trivially satisfied at each intermediate node. We assume the following robust model: When some arc fails, all flow routed on paths containing that arc cannot reach the sink. In other words, the flow units are not allowed to change their routes in the intermediate nodes, i.e., each flow unit must take the way that has been decided in advance. In the adaptive model, one is allowed to reoptimize the routing of the flow onto different paths after the adversary has selected the set of failing arcs.

Given an $s$-t-flow $x$ with path decomposition $(x_P)_{P \in P \cup C}$, we define the robust value $RVal(x)$ as the amount of flow reaching the sink in the worst case, that is,
\[
RVal(x) := \min_{P \in \mathcal{P}} \sum_{e \in P} (1 - \lambda_e) x_P
\]
\[
\text{s.t. } \sum_{e \in E} \mu_e \leq \Gamma,
\]
\[
\lambda_e - \sum_{e \in P} \mu_e \leq 0 \quad \forall P \in \mathcal{P},
\]
\[
\mu_e \in \{0, 1\} \quad \forall e \in E,
\]
\[
\lambda_e \in \{0, 1\} \quad \forall P \in \mathcal{P}.
\]
Here, the variable \( \mu_e \) is either 1 or 0, indicating whether or not arc \( e \) fails. The first constraint controls the number of arc failures. The variable \( \lambda_P \) is either 1 or 0, indicating whether or not path \( P \) fails. Thus, the objective function gives the amount of flow which is able to reach the sink. For each \( P \in \mathcal{P} \), the variable \( \lambda_P \) appears in the objective function with \(-x_P\). Because this is negative, in an optimum solution \( \lambda_P \) will always be set to 1 if possible. The second set of constraints ensures that this can be done, only if some arc \( e \in P \) fails.

Notice that in Problem (33) \( \lambda_P \) for each path \( P \in \mathcal{P} \) will be as large as possible in an optimum solution. Therefore, if we know optimal values for the variables \( \mu_e \), we can get optimal values for \( \lambda \) by

\[
\lambda_P = \min \left\{ 1, \sum_{e \in P} \mu_e \right\}, \quad \forall P \in \mathcal{P}.
\]

Thus, Problem (33) can be rewritten as follows:

\[
\text{RVal}(x) := \min \sum_{P \in \mathcal{P}} \left( 1 - \min \left\{ 1, \sum_{e \in P} \mu_e \right\} \right) x_P \\
\text{s.t.} \sum_{e \in E} \mu_e \leq \Gamma, \\
\mu_e \in \{0, 1\}, \quad \forall e \in E.
\]

Although the extreme points of the feasible region of Problem (35) are all binary, the objective function is convex and hence it does not attain its minimum at an extreme point in general. Therefore, there is an integrality gap, and by relaxing the binary variable \( \mu \) to be continuous, one would incur a loss in the optimal value in general. As an example, consider the path-based flow \( x^1 \) in Figure 6(c). It is easy to check that the fractional optimal solution of Problem (35) is \( \mu_{(e, v_1)} = \mu_{(e, v_6)} = \mu_{(e, v_7)} = 0.5 \) and \( \mu_e = 0 \) for all other arcs \( e \). Its objective function value is 2. The optimal integral solution is \( \mu_{(e, v_1)} = \mu_{(e, v_7)} = 1 \) and \( \mu_e = 0 \) for all other arcs \( e \). This has objective function value 2.5.

For the special case \( \Gamma = 1 \), we must have \( \lambda_P = \sum_{e \in P} \mu_e \) for each \( P \in \mathcal{P} \) in an optimal solution. Therefore, we can write

\[
\sum_{P \in \mathcal{P}} \lambda_P x_P = \sum_{P \in \mathcal{P}} \left( \sum_{e \in P} \mu_e \right) x_P = \sum_{e \in E} \mu_e \left( \sum_{P \in \mathcal{P}, e \in P} x_P \right) = \sum_{e \in E} \mu_e x_e
\]

and simplify Problem (33) as follows:

\[
\text{RVal}(x) := \sum_{P \in \mathcal{P}} x_P - \max_{e \in E} x_e.
\]

This implies that the robust value of \( x \) is independent of a particular path decomposition of \( x \) for \( \Gamma = 1 \).

We next show that this result does not hold in general, already for \( \Gamma = 2 \). Consider the network shown in Figure 6 with 9 nodes and 15 arcs. The numbers on the arcs indicate the flow values \( x \). Assume that two arcs can fail. The adaptive value of this flow is 3. Now let us compute the robust value of \( x \) based on the path formulation. Let \( (x^p_e)_{P \in \mathcal{P}} \) and \( (x^p_e)_{P \in \mathcal{P}} \) be two path decompositions of \( x \) as shown in Figures 6(b) and 6(c), respectively. The robust value of \( x^1 \) is 2 and the robust value of \( x^2 \) is 8/3. This proves that the robust value of an (arc-based) flow \( x \) depends on the path decomposition. Therefore, in this section, when referring to a flow \( x \) we mean a particular path decomposition of \( x \).
If the path decomposition of a flow must remain fixed, the solutions becomes more vulnerable to arc failures.

We now define the problem to find a path-based flow $x$ with maximum robust value:

**Robust Maximum Flow Problem in Path-Based Formulation**

Input: A network $N = (V, E)$ with a source $s \in V$, a sink $t \in V$, arc capacities $u_e \in \mathbb{R}_+$ for $e \in E$ such that at most $\Gamma$ arcs can fail.

Task: Find a flow in a path-based formulation whose robust value is maximum.

This problem can be formulated as follows:

$$\max \sum_{e \in P} x_e - \max_{\lambda \in \Lambda} \sum_{P \in \mathcal{P}} \lambda_P x_P$$

s.t. constraints of Problem (33),

where $\mathcal{P}$ is the set of all $s,t$-flows in path-based formulation, i.e.,

$$\mathcal{P} := \left\{ x = (x_P)_{P \in \mathcal{P}} \mid \sum_{P \in \mathcal{P}, e \in P} x_e \leq u_e \text{ for each } e \in E \right\}.$$

### 4.1. Approximation Results

It follows from Problem (36) that the problem of finding a flow with maximum robust value is solvable in polynomial time for the special case of $\Gamma = 1$. This result has already been observed by Aneja et al. (2001), who consider the case where only one arc is permitted to fail. Later, Du and Chandrasekaran (2007) present an alternative but equivalent formulation of the robust maximum flow problem in path-based formulation for the case where two arcs can fail. They show that the problem is NP-hard. Hence, we attempt to design approximation algorithms to obtain near-optimal solutions by considering the LP relaxation of Problem (33).

For a given flow $x$, we define $z(x)$ to be the optimal value of the following 0-1 integer optimization problem:

$$z(x) := \max \sum_{P \in \mathcal{P}} \lambda_P x_P$$

s.t. constraints of Problem (33).

This problem can be seen as an instance of the budgeted maximum coverage problem:

**Budgeted Maximum Coverage Problem**

Input: A ground set $U$ of $l$ elements with associated weights $\{w_i\}_{i=1}^l$, a set of subsets $\mathcal{B} = \{B_1, \ldots, B_k\}$ such that for $1 \leq i \leq k$, $B_i \subseteq U$, and a parameter $\Gamma \leq k$.

Task: Find a subset $\mathcal{B} \subseteq \mathcal{B}$ with $|\mathcal{B}| \leq \Gamma$ such that the total weight of elements covered by $\mathcal{B}$ is maximized.

Problem (38) reduces to the budgeted maximum coverage problem by letting $U := \{P \in \mathcal{P} \mid x_P > 0\}$ with $w_P = x_P$ for each path $P \in U$ and $\mathcal{B} := \{B_e \mid e \in E\}$ with $B_e = \{P \in \mathcal{P} \mid e \in P\}$ for each arc $e \in E$. The budgeted maximum coverage problem is known to be NP-hard (see, e.g., Hochbaum 1997). Ageev and Sviridenko (1999) show the LP relaxation yields a solution whose value is within a factor of $1 - (1 - 1/r)^\Gamma$ of the optimal value, where $r$ is the maximum size of the subsets. In our problem, we have $r \leq n - 1$ because each path contains at most $n - 1$ arcs. Suppose that $f(x)$ denotes the optimal value of the LP relaxation of Problem (38). Then, we can conclude $pf(x) \leq z(x) \leq f(x)$.

Herein and throughout the rest of this section, $\rho$ denotes the value $(1 - (1 - 1/n)^\Gamma)$.}

**Theorem 5.** Let $x^*$ be a flow with maximum robust value and $\tilde{x}, \tilde{\delta}$ be an optimal solution of the following LO problem:

$$\max \sum_{e \in B^\delta(s)} x_e - \sum_{e \in B^\delta(t)} x_e - \Gamma \tilde{\delta}$$

s.t. constraints of Problem (39),

$$0 \leq x_e \leq u_e \forall e \in E,$$

$$x_e \leq \tilde{\delta} \forall e \in E.$$

Then,

$$0 \leq RVal(x^*) - RVal(\tilde{x}) \leq \frac{1 - \rho}{\rho} (Val(x^*) - RVal(x^*)) \leq (1 - (1 - 1/n)^\Gamma).$$

Problem (39) gives us the flow $\tilde{x}$ in an arc-based formulation. Thus, in order to prove Theorem 5, we need to show the following lemma: If $\tilde{x}, \tilde{\delta}$ is an optimal solution for Problem (39), then the robust value of $\tilde{x}$ is $Val(\tilde{x}) - \Gamma \tilde{\delta}$ for any path decomposition of $\tilde{x}$.

**Lemma 9.** Suppose that $\tilde{x}, \tilde{\delta}$ is an optimal solution for Problem (39) and $(\tilde{x}_P)_{P \in \mathcal{P}}$ is an arbitrary path decomposition of $\tilde{x}$. There exist $\Gamma$ arcs, say $e_1, \ldots, e_{\Gamma}$, with $\tilde{x}_{e_1} = \cdots = \tilde{x}_{e_{\Gamma}} = \tilde{\delta}$ such that at most one of them lies on each path $P$ with $\tilde{x}_P > 0$.

**Proof.** If the optimal value of Problem (39) is zero, then $\tilde{x} = 0$ and $\tilde{\delta} = 0$ is the unique optimal solution, and the lemma is straightforward. Therefore, we assume that the optimal value of Problem (39) is strictly positive.

Suppose, by contradiction, that there are at most $k$ ($k < \Gamma$) arcs $e_1, \ldots, e_k$ with $\tilde{x}_{e_1} = \cdots = \tilde{x}_{e_k} = \tilde{\delta}$ such that each path $P$ with $\tilde{x}_P > 0$ contains at most one of these arcs. This implies that there are $k$ flow-carrying paths $P_1, \ldots, P_k$ such that each arc $e$ with $\tilde{x}_e = \tilde{\delta}$ lies on one of these paths. We now define a new solution $x, \delta$ for Problem (39) by setting $\delta := \tilde{\delta} - \epsilon$ and $x_P := \tilde{x}_P - \epsilon$, if $P = P_1, \ldots, P_k$, and $x_P := \tilde{x}_P$, otherwise. The $\epsilon$ is strictly positive and is chosen small enough to ensure that $\sum_{P \in \mathcal{P}, e \in P} \tilde{x}_P \leq \delta$. The objective function value of Problem (39) for $x, \delta$ is

$$\sum_{P \in \mathcal{P}} x_P - \Gamma \delta = \sum_{P \in \mathcal{P} \setminus \{P_1, \ldots, P_k\}} x_P + \sum_{i=1}^k x_P - \Gamma \delta$$

$$= \sum_{P \in \mathcal{P} \setminus \{P_1, \ldots, P_k\}} \tilde{x}_P + \sum_{i=1}^k (\tilde{x}_P - \epsilon) - \Gamma (\tilde{\delta} - \epsilon)$$

$$= \sum_{P \in \mathcal{P}} \tilde{x}_P - \Gamma \tilde{\delta} + (\Gamma - k) \epsilon > \sum_{P \in \mathcal{P}} \tilde{x}_P - \Gamma \tilde{\delta}.$$
This contradicts with the optimality of $\bar{x}$, $\bar{\delta}$, which proves the lemma. □

**Proof of Theorem 5.** We first notice that for each $e \in E$, the upper-bound constraint $\mu_e \leq 1$ is redundant (as long as $\lambda_p \leq 1$), since assigning a larger value to $\mu_e$ does not contribute to the objective function value. We now consider the LP relaxation of Problem (33) and let $g(x)$ denote its optimal value. Considering the dual problem, we can express $g(x)$ as follows

$$g(x) = \max \sum_{P \in \mathcal{P}} x_P - \Gamma \delta - \sum_{P \in \mathcal{P}} \beta_P$$

s.t. $\delta - \sum_{P \in \mathcal{P}: e \in P} \alpha_p \geq 0 \quad \forall e \in E,$

$$\alpha_p + \beta_P \geq x_P \quad \forall P \in \mathcal{P},$$

$$\alpha_p, \beta_P \geq 0 \quad \forall P \in \mathcal{P}.$$ (40)

In this problem, if we know the optimal values for $\alpha$ and $\delta$, then the upper-bound value for $\beta$ can be computed by $\beta_p := \max(0, x_p - \alpha_p)$ for each $P \in \mathcal{P}$. Moreover, we can assume $\alpha_p \leq x_p$ for each $P \in \mathcal{P}$, because if $\alpha_p > x_p$ for some path $P$, then reducing the value $\alpha_p$ to $x_p$ yields a feasible solution for Problem (40), whose objective function value does not decrease. This implies that

$$g(x) = \max \sum_{P \in \mathcal{P}} \alpha_p - \Gamma \delta$$

s.t. $\delta - \sum_{P \in \mathcal{P}: e \in P} \alpha_p \geq 0 \quad \forall e \in E,$

$$0 \leq \alpha_p \leq x_p \quad \forall P \in \mathcal{P}.$$ (41)

We proceed to show that

$$g(x) = \max \sum_{e \in \mathcal{E}^+} \alpha_e - \sum_{e \in \mathcal{E}^-} \alpha_e - \Gamma \delta$$

s.t. $\sum_{e \in \mathcal{E}^+} \alpha_e - \sum_{e \in \mathcal{E}^-} \alpha_e = 0 \quad \forall v \in V \setminus \{s, t\},$

$$\alpha_e \leq \delta \quad \forall e \in E,$$

$$0 \leq \alpha_e \leq x_e \quad \forall e \in E,$$ (42)

where $x_e := \sum_{P \in \mathcal{P}: e \in P} x_P$ for each arc $e \in E$. Let $\alpha$ and $\delta$ be a feasible solution for Problem (41). Then it is not hard to see that $\alpha_e := \sum_{P \in \mathcal{P}: e \in P} \alpha_p, e \in E$ and $\delta$ yields a feasible solution for Problem (42) with the same objective value. Conversely, suppose that $(\alpha_e)_{e \in E}$ and $\delta$ is a feasible solution for Problem (42). As $\alpha_e \leq x_e$ for each $e \in E$, there is a path decomposition $(\rho_P)_{P \in \mathcal{P}}$ of $(\alpha_e)_{e \in E}$ so that $\alpha_p \leq x_p$ for each $P \in \mathcal{P}$. This implies that $(\rho_P)_{P \in \mathcal{P}}$ and $\delta$ is a feasible solution for Problem (41), whose objective function value equals $\sum_{P \in \mathcal{P}} \alpha_p - \Gamma \delta$.

Summarizing the above discussion, the problem of finding a flow $x$ with maximum value $f(x)$ reads as follows:

$$\max \sum_{e \in \mathcal{E}^+} \alpha_e - \sum_{e \in \mathcal{E}^-} \alpha_e - \Gamma \delta$$

s.t. $\sum_{e \in \mathcal{E}^+} \alpha_e - \sum_{e \in \mathcal{E}^-} \alpha_e = 0 \quad \forall v \in V \setminus \{s, t\},$

$$\sum_{e \in \mathcal{E}^+} \alpha_e - \sum_{e \in \mathcal{E}^-} \alpha_e = 0 \quad \forall v \in V \setminus \{s, t\},$$ (43)

$$0 \leq \alpha_e \leq x_e \quad \forall e \in E,$$

$$0 \leq \alpha_e \leq \delta \quad \forall e \in E.$$

Clearly, the above problem has an optimal solution with the property that $x_e = \alpha_e$ for each $e \in E$. This shows that Problem (43) is equivalent to Problem (39) if the binary variables $\mu_e$ for all $e \in E$ are relaxed to be continuous.

We know by Lemma 9 that the robust value of $\tilde{x}$ is $\text{Val}(\tilde{x}) - \Gamma \delta$ for any path decomposition of $\tilde{x}$. We therefore consider an arbitrary path decomposition $(\tilde{x}_p)_{P \in \mathcal{P}}$ of $\tilde{x}$. We next let $z(x^*)$ be the optimal value of Problem (38) with respect to flow $x^*$, and $f(x^*)$ be the optimal value of the LP-relaxation of Problem (38). Then,

$$\text{RVal}(\tilde{x}) = \text{Val}(\tilde{x}) - \Gamma \delta \geq \text{Val}(x^*) - f(x^*)$$

$$\geq \text{Val}(x^*) - \frac{1}{\rho} \text{Val}(x^*) = \frac{1}{\rho} \text{RVal}(x^*) - \frac{1}{\rho} \text{Val}(x^*).$$

The first inequality follows from the optimality of the solution $\tilde{x}$, $\Gamma$ for Problem (39). The second inequality follows from the fact that the LP relaxation of Problem (38) yields a $p$-approximation, and the last equality is trivial. Hence, we can write

$$0 \leq \text{RVal}(x^*) - \text{RVal}(\tilde{x})$$

$$\leq \text{RVal}(x^*) - \frac{1}{\rho} \text{RVal}(x^*) + \frac{1}{\rho} \text{Val}(x^*)$$

$$= \frac{1}{\rho} (\text{Val}(x^*) - \text{RVal}(x^*)).$$ □

Theorem 5 implies that if there is a flow $x^*$ with robust value larger than $\beta$ times its nominal value for some $0 < \beta \leq 1$, then Problem (39) yields a $1 - ((1 - \rho)/\beta) \cdot ((1 - \beta)/\beta)$-approximation. The larger $\beta$, the better the performance guarantee.

**Corollary 2.** Let $x^*$ be a flow with maximum robust value, such that $\beta \text{Val}(x^*) \leq \text{RVal}(x^*)$ for some $\beta \in (0, 1]$. Further, suppose that $\tilde{x}$, $\tilde{\delta}$ is an optimal solution for Problem (39). Then, $\tilde{x}$ is a $1 - ((1 - \rho)/\beta)((1 - \beta)/\beta)$-approximation, i.e.,

$$\text{RVal}(\tilde{x}) \geq \left(1 - \frac{1 - \rho}{\beta} \frac{1 - \beta}{\beta}\right) \text{RVal}(x^*).$$

In particular, if $\beta \geq 1/2$, then $\tilde{x}$ is an $\alpha$-approximation for some $\alpha \geq 0.58197$. 
The following result shows that Problem (39) can also be used to obtain approximative solutions for the adaptive maximum flow problem.

**Corollary 3.** Let $x^*$ be an (arc-based) adaptive maximum flow and $\tilde{x}, \tilde{\delta}$ be an optimal solution for Problem (39). Suppose that there exists a path decomposition of $x^*$, say $(x^*_{p_{ij}})_{p_{ij}}$, such that $\beta \Val(x^*) \leq \Val(x^*)$ for some $\beta \in (0, 1]$. Then, $\tilde{x}$ yields a $\beta(1 - (((1 - \rho)/\rho) \cdot ((1 - \beta)/\beta)))$-approximation for the adaptive maximum flow problem, that is,

$$AVal(\tilde{x}) \geq \beta \left(1 - \frac{1 - \rho}{\rho} \frac{1 - \beta}{\beta}\right) AVal(x^*).$$

**Proof.** From Lemma 9 we get that $AVal(\tilde{x}) = Val(\tilde{x}) - \Gamma \tilde{\delta}$. The result now follows from Corollary 2 as

$$AVal(\tilde{x}) = Val(\tilde{x}) - \Gamma \tilde{\delta} \geq \left(1 - \frac{1 - \rho}{\rho} \frac{1 - \beta}{\beta}\right) RVal(x^*)$$

$$\geq \beta \left(1 - \frac{1 - \rho}{\rho} \frac{1 - \beta}{\beta}\right) AVal(x^*).$$

So far we have confined our attention on maximizing the robust value of a flow. However, it might be the case that two flows with the same robust value have different nominal values. Here it is reasonable to prefer the flow whose nominal value is better. In other words, one may be interested in finding flows with maximum robust and nominal values. The problem then becomes a biobjective optimization problem, and in general there is no solution that simultaneously maximizes both objective functions (see the example in Figure 4). This motivates us to seek a flow $x$ so that the sum of the nominal value and the robust value is maximized. We define the total value $TVal(x)$ of a flow $x$ as

$$TVal(x) := Val(x) + RVal(x).$$

**Theorem 6.** Let $x^*$ be an $s$-$t$-flow with maximum total value and $\tilde{x}, \tilde{\delta}$ be an optimal solution of the following LO model:

$$\text{max } 2 \sum_{e \in \delta^+(s)} x_e - 2 \sum_{e \in \delta^-(s)} x_e - \Gamma \tilde{\delta}$$

$$\text{s.t. } \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s, t\},$$

$$0 \leq x_e \leq u_e \quad \forall e \in E,$$

$$x_e \leq \tilde{\delta} \quad \forall e \in E.$$

Then,

$$0 \leq TVal(x^*) - TVal(\tilde{x}) \leq \frac{1}{\rho} (Val(x^*) - RVal(x^*)).$$

**Proof.** Let $z(x^*)$ be the optimal value of Problem (38) and $f(x^*)$ be the optimal value of the LP relaxation of Problem (38) with respect to $x^*$. Then,

$$TVal(\tilde{x}) = Val(\tilde{x}) + RVal(\tilde{x}) = 2Val(\tilde{x}) - \Gamma \tilde{\delta}$$

$$\geq 2Val(x^*) - f(x^*)$$

$$\geq 2Val(x^*) - \frac{1}{\rho} z(x^*)$$

$$\geq \frac{1}{\rho} RVal(x^*) + \frac{2\rho - 1}{\rho} Val(x^*).$$

Here, the first inequality follows from the optimality of the solution $\tilde{x}, \Gamma$ for Problem (44), the second inequality follows from fact that the LP-relaxation of Problem (38) yields a $\rho$-approximation, and the last equality is trivial. Thus, we can write

$$0 \leq TVal(x^*) - TVal(\tilde{x}) = Val(x^*) + RVal(\tilde{x}) - \frac{1}{\rho} RVal(x^*) - \frac{2\rho - 1}{\rho} Val(x^*)$$

$$\leq \frac{1}{\rho} (Val(x^*) - RVal(x^*)).$$

**Corollary 4.** Let $x^*$ be an $s$-$t$-flow with maximum total value and $\tilde{x}, \tilde{\delta}$ be an optimal solution for Problem (44). Suppose that $((1 - \beta)/(1 + \beta)) Val(x^*) \leq RVal(x^*)$ for some $\beta \in (0, 1]$. Then, $\tilde{x}$ is an $1 - (((1 - \rho)/\rho)(1/\beta))$-approximation, i.e.,

$$TVal(\tilde{x}) \geq \left(1 - \frac{1 - \rho}{\rho} \frac{1}{\beta}\right) TVal(x^*).$$

Notice that $((1 - \beta)/(1 + \beta)) Val(x^*) \leq RVal(x^*)$ for $\beta = 1$. Hence, $\tilde{x}$ is an $\alpha$-approximation with $\alpha \geq 0.28171$.

**5. Adaptive Maximum Flows Over Time**

Thus far we have considered the setting of static network flow problems in which time does not enter into the model, or the time to traverse an arc is assumed to be zero. However, temporal dynamics is an important feature in many practical applications of network flow problems. To capture these situations, Ford and Fulkerson (1958, 1962) introduce the notion of flows over time (also called dynamic flows in the literature). In contrast to classical static flows in the dynamic model each arc has transit time, describing how long it takes to traverse that arc. In the dynamic model the flow on an arc is not a constant value but a function over time. Ford and Fulkerson (1958, 1962) study the maximum flow over time problem where the aim is to find the maximum amount of flow that can be sent from a source node to a sink node within a given time horizon. They show that this problem can be solved efficiently by one minimum cost flow computation on the given network. Since then, many authors have extensively studied different features of flows over time (see, e.g., Aronson 1989, Powell et al. 1995, Skutella 2009, and the references therein).

In the rest of this section, we study a generalization of the adaptive maximum flow problem to flows over time. Each arc $e \in E$ has an associated transit time $\tau_e \in \mathbb{Z}_+$ specifying the required amount of time for traveling from the tail to the head of $e$; hence, if flow leaves node $v$ at time $\theta$, it arrives at $w$ at time $\theta + \tau_e$. For a given time horizon $T \in \mathbb{Z}_+$, a flow over time is a Lebesgue-integrable function

$$f_\cdot : [0, T) \rightarrow \mathbb{R}_+ \quad \forall e \in E,$$
where the value \( f_e(\theta) \) represents the rate at which flow enters arc \( e \) at time \( \theta \).

In this setting, the arc capacity \( u_e \) limits the inflow rate, i.e., the amount of flow that can enter arc \( e \) per time unit. Formally, we require that a flow over time \( f \) fulfills the following arc capacity constraints:

\[
f_e(\theta) \leq u_e \quad \forall e \in E, \theta \in [0, T).
\]

The flow over time \( f \) induces a storage function \( \text{ex}_f \) on \( [0, T) \) at each node \( v \in V \) by the following flow conservation constraint

\[
\text{ex}_f(v, \theta) := \sum_{e \in \delta^+(v)} \int_0^\theta f_e(\theta) \, d\theta - \sum_{e \in \delta^-(v)} \int_0^{\theta - \tau_e} f_e(\theta) \, d\theta.
\]

Here, the first sum represents the total amount of flow arriving at node \( v \) up to time \( \theta \), and the second sum represents the total amount of flow leaving node \( v \) up to time \( \theta \). Thus, the value \( \text{ex}_f(v, \theta) \) gives the net amount of flow that enters node \( v \) up to time \( \theta \).

A flow \( f \) is called an \( s-t \)-flow over time, if \( \text{ex}_f(v, \theta) \geq 0 \), for each \( v \in V \setminus \{s\} \) and \( \theta \in [0, T) \), and \( \text{ex}_f(s, T) = 0 \), for each \( v \in V \setminus \{s, t\} \). The value of \( f \) is the total net flow into the sink node \( t \) up to time \( T \), i.e., \( \text{Val}(f) := \text{ex}_f(t, T) \).

In the maximum flow over time problem we seek an \( s-t \)-flow over time with maximum value. This problem can be formulated as follows:

\[
\begin{align*}
\text{max} & \quad \text{Val}(f) \\
\text{s.t.} & \quad \text{ex}_f(v, \theta) \geq 0 \quad \forall v \in V \setminus \{s\}, \theta \in [0, T), \\
& \quad \text{ex}_f(s, T) = 0 \quad \forall v \in V \setminus \{s, t\}, \\
& \quad 0 \leq f_e(\theta) \leq u_e \quad \forall e \in E, \theta \in [0, T).
\end{align*}
\]

Now we again assume that arcs can fail. As in the static case, suppose that at most \( \Gamma \) arcs will fail. Whenever an arc fails, it disappears for the entire time horizon. With respect to a flow \( f \) and a scenario of arc failures \( \mu \in \Lambda \), we define \( M(f, \mu) \) as the value of the maximum flow over time that can be pushed through the failing network, i.e.,

\[
M(f, \mu) := \max \text{Val}(g)
\]

\[
\begin{align*}
\text{s.t.} & \quad \text{ex}_g(v, \theta) \geq 0 \\
& \quad \forall v \in V \setminus \{s\}, \theta \in [0, T), \\
& \quad \text{ex}_g(s, T) = 0 \quad \forall v \in V \setminus \{s, t\}, \\
& \quad 0 \leq g_e(\theta) \leq (1 - \mu_e) f_e(\theta) \\
& \quad \forall e \in E, \theta \in [0, T).
\end{align*}
\]

Now \( \text{AVal}(f) := \min_{\mu \in \Lambda} M(f, \mu) \) is called the adaptive value of \( f \). We say \( f \) is an adaptive maximum flow over time if it has the maximum adaptive value among all \( s-t \)-flows over time. We summarize the description of this problem.

**Adaptive Maximum Flow Over Time Problem**

**Input:** A network \( G = (V, E) \) with source \( s \in V \), sink \( t \in V \), arc capacities \( u_e \in \mathbb{R}_+ \), \( e \in E \), transit times \( \tau_e \in \mathbb{Z}_+ \), \( e \in E \), and time horizon \( T \in \mathbb{Z}_+ \) such that at most \( \Gamma \) arcs can fail.

**Task:** Find an adaptive maximum flow over time and its adaptive value.

### 5.1. Lack of the Temporally Repeated Flow Property

A nice feature of the maximum flow over time problem is that there is always an optimal solution of a particularly simple kind: a so-called temporally repeated flow, which is induced by a static \( s-t \)-flow. Given a static \( s-t \)-flow with path decomposition \( x = (x_P)_{P \in \mathcal{P}} \), the corresponding temporally repeated flow \( f^* \) is obtained as follows: start sending flow at rate \( x_P \) into path \( P \) from the source \( s \) during the time interval \([0, T - \tau_P]\) and let the flow proceed towards the sink without any delay at intermediate nodes (see Skutella 2009 for more details). It is a well-known result by Ford and Fulkerson (1958, 1962) that temporally repeated flows are sufficient to solve the maximum flow over time problem, and such a flow can be obtained by one minimum cost flow computation on the given network, where transit times of arcs are interpreted as arc costs. Unlike the deterministic case, the following example shows that the adaptive variant of the maximum flow over time problem in general cannot be solved when restricting to temporally repeated flows.

Consider the network depicted in Figure 7. Denote the arcs by \( e_1 = (s, u) \), \( e_2 = (s, w) \), \( e_3 = (u, v) \), \( e_4 = (w, v) \), \( e_5 = (u, t) \), \( e_6 = (w, t) \), and \( e_7 = (v, t) \) and the paths by \( P_1 := s, u, t \), \( P_2 := s, w, v, t \), \( P_3 := s, w, v, t \), and \( P_4 := s, w, t \). Each arc has unit capacity, and the transit times are shown on the arcs in Figure 7. Set the time horizon to \( T := 6 \). The maximum flow over time value is \( 5 \). There is a unique maximum temporally repeated flow over time \( f \) given as follows: send flow at rate \( 1 \) into paths \( P_2 \) and \( P_4 \), respectively, during the time intervals \([0, 4]\) and \([0, 1]\). There is also another maximum flow over time \( f^* \), which is not a temporally repeated flow over time, given by

\[
f'_{e_1}(\theta) = \begin{cases} 1 & \text{for } \theta \in [0, 3), \\ 0 & \text{for } \theta \not\in [0, 3), \\ \end{cases} \quad f'_{e_2}(\theta) = \begin{cases} 1 & \text{for } \theta \in [0, 1) \cup [2, 3), \\ 0 & \text{for } \theta \not\in [0, 1) \cup [2, 3), \\ \end{cases} \quad f'_{e_3}(\theta) = \begin{cases} 1 & \text{for } \theta \in [2, 4), \\ 0 & \text{for } \theta \not\in [2, 4), \\ \end{cases} \quad f'_{e_4}(\theta) = \begin{cases} 1 & \text{for } \theta \in [3, 4), \\ 0 & \text{for } \theta \not\in [3, 4), \\ \end{cases} \quad f'_{e_5}(\theta) = \begin{cases} 1 & \text{for } \theta \in [2, 5), \\ 0 & \text{for } \theta \not\in [2, 5), \\ \end{cases} \quad f'_{e_6}(\theta) = \begin{cases} 1 & \text{for } \theta \in [1, 2), \\ 0 & \text{for } \theta \not\in [1, 2), \\ \end{cases} \quad f'_{e_7}(\theta) = \begin{cases} 1 & \text{for } \theta \in [2, 5), \\ 0 & \text{for } \theta \not\in [2, 5), \\ \end{cases}
\]
Theorem 7. The adaptive maximum flow over time problem is weakly NP-hard.

Proof. We show hardness for the related decision problem that asks whether there is an s-t-flow over time with adaptive value strictly larger than zero. We reduce the well-known weakly NP-complete partition problem to this problem.

**Partition Problem**

- **Input:** n integers $a_1, \ldots, a_n \in \mathbb{N}$ with $\sum_{i=1}^{n} a_i = 2L$ for some $L \in \mathbb{N}$.
- **Task:** Is there a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} a_i = L$?

The reduction is similar to that of Melkonian (2007), who proves that the maximum flow over time problem with aggregate arc capacities is weakly NP-hard. Given an instance of the partition problem, we construct a network as follows: Consider a series-parallel network, as shown in Figure 8, with node set $\{1, \ldots, n+1\}$; let $s = 1$ and $t = n$. For each $i = 1, \ldots, n$, there are two parallel arcs $e_i$ and $e_i'$ linking node $i$ to node $i+1$. We assign a unit capacity to all arcs. We let $E := \{e_1, \ldots, e_n\}$ and $E' := \{e_1', \ldots, e_n'\}$ and assign transit times to the arcs as follows: $\tau_{e_i} := na_{\text{max}} a_i$ for $e_i \in E$ and $\tau_{e_i'} := (na_{\text{max}} + 1)a_i$ for $e_i' \in E'$, where $a_{\text{max}} := \max\{a_1, \ldots, a_n\}$. Let $T := (2a_{\text{max}}n + 1)L + 1$ and $\Gamma = 1$.

We have to show that the partition problem is a “yes” instance, if and only if there exists an adaptive maximum flow over time whose adaptive value is strictly larger than zero.

First assume that the partition problem is a “yes” instance and let $I$ be a subset of $\{1, \ldots, n\}$ with $\sum_{i \in I} a_i = L$. Now consider two arc-disjoint paths $P_1$ and $P_2$ from $s$ to $t$ as follows. Path $P_1$ contains the arcs $e_i \in E$ with $i \in I$ and the arcs $e_i' \in E'$ with $i \in I := \{1, \ldots, n\}\backslash I$. Path $P_2$ contains the remaining arcs of the network. The transit time $\tau_{P_1}$ of path $P_1$ is the sum of the transit times of the arcs in $P_1$:

$$\tau_{P_1} = \sum_{e_i \in E \cap I} \tau_{e_i} + \sum_{e_i' \in E' \cap I} \tau_{e_i'} = na_{\text{max}} L + (na_{\text{max}} + 1)L = (2a_{\text{max}}n + 1)L.$$

Similarly, we have $\tau_{P_2} = (2a_{\text{max}}n + 1)L$.

Now consider a flow over time $f$ by sending a flow at rate one into paths $P_1$ and $P_2$ during the time interval $[0, 1]$. It is clear that the adaptive value of $f$ is 1 since when one arc fails (no matter which one), then one unit of flow reaches the sink. This completes the proof in one direction.

We now assume that the adaptive maximum flow over time value is strictly positive. This implies the existence of two arc-disjoint paths $P_1$ and $P_2$, whose transit times are at most $T - 1$. We now define $I := \{i \in \{1, \ldots, n\} \mid e_i \in E \text{ on path } P_1\}$. We show that $\sum_{i \in I} a_i = L$. Assume that $\sum_{i \in I} a_i = L + \delta$ for some nonzero $\delta$. Then,

$$\tau_{P_1} = \sum_{e_i \in E \cap I} \tau_{e_i} + \sum_{e_i' \in E' \cap I} \tau_{e_i'} = na_{\text{max}} (L + \delta) + (na_{\text{max}} + 1)(L - \delta) = (2a_{\text{max}}n + 1)L - \delta = T - 1 - \delta,$$

and

$$\tau_{P_2} = \sum_{e_i \in E' \cap I} \tau_{e_i} + \sum_{e_i' \in E \cap I} \tau_{e_i'} = na_{\text{max}} (L - \delta) + (na_{\text{max}} + 1)(L + \delta) = (2a_{\text{max}}n + 1)L + \delta = T - 1 + \delta.$$

This contradicts the fact that the transit times of $P_1$ and $P_2$ are at most $T - 1$. Thus, we must have $\sum_{i \in I} a_i = L$. $\Box$
6. Computational Results

The computational study in this section serves a double purpose. First, we want to validate the robust models, in particular, the adaptive maximum flow problem. Second, this computational study tests the ability of the proposed LO-method to compute good adaptive flows on instances of relevant size.

Because the adaptive flow problem is computationally hard, we will, on the one hand, give an upper bound on the maximum adaptive flow value, and, on the other hand, report the exact adaptive flow value of solutions found by the best of our approaches, which is here the LO Program (39). As an upper bound we use the optimal network interdiction value of the same network for which we construct the adjustable flow. Note that the optimal adaptive flow value must be less than the network’s interdiction value. We assess the quality of our heuristically found flows by their gap to the interdiction value.

To validate our model of adaptive flows we show that these flows perform significantly better in an adverse environment than other, possibly simpler, concepts. To this end, we compare heuristically good, adaptive flows to two simpler competitors. Firstly, we compare the adaptive value of the heuristically found flow to the adaptive value of a maximum flow. This comparison shows that if an adverse scenario occurs, it pays to have used (even only heuristically found) adaptive flows instead of maximum flows. The results show that one can significantly improve the worst-case value of a flow, when one considers the adverse scenarios already during planning. Secondly, we compare the heuristically found adaptive flows to an optimal (nonadjustable) robust flow. The (nonadjustable) robust maximum flow value is computed easily even for large \( \Gamma \). However, its value vanishes already for very small values of \( \Gamma \). Thus, it is overly conservative to use nonadjustable robust flows as a mathematical model, if the application allows for post-failure adjustments.

All computations have been carried out on a 2.13 GHz Intel Core 2 Duo processor with 2 GB RAM and a time limit of one hour. As modeler and LO solver, we used the components of the free GLPK (see http://www.gnu.org/s/glpk/) software package.

We test two types of instances. One part of the instance set are NETGEN benchmark instances developed by Klingman et al. (1974). From these we picked 10 instances with a sufficiently high number of source nodes. We then connected the source nodes to a super source (and likewise the sink nodes to a super sink). We exclude from failure those arcs that lead from the super source to the super nodes. This, together with the high number of source nodes, is in favor of the (nonadjustable) robust flow model, which trivially equals zero for any positive \( \Gamma \), e.g., if all outgoing arcs of the source can fail and none of them are parallel.

The computational results on these instances are reported in Table 1. The first two columns report the size of the network. In columns RF (and ARVal) we report the smallest value of \( \Gamma \) for which a robust flow (respectively, an adaptive flow) vanishes. The column headed ARVal/Ubd displays the average ratio of the adaptive flow value found by Problem (39) to the upper bound given by the network interdiction problem. By MF/ARVal we denote the average ratio of the adaptive value of a maximum flow to the adaptive flow value found by Problem (39). In both cases, the average is taken over all choices of \( \Gamma \) with nonvanishing flow values.

We observe that despite the careful construction of the instances, the robust flow value vanishes already for small values of \( \Gamma \) (between 1 and 4), whereas the adaptive flow is zero only for high \( \Gamma \) values (between 17 and 69).

We see that the heuristically found adaptive flows are always close to the upper bounds, whereas a maximum flow for the nominal problem is up to 40\% worse (percentages averaged over all nonvanishing choices of \( \Gamma \)).

### Table 1.

<table>
<thead>
<tr>
<th>Instance</th>
<th># nodes</th>
<th># arcs</th>
<th>RF</th>
<th>ARVal</th>
<th>ARVal/Ubd (%)</th>
<th>MF/ARVal (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>401</td>
<td>1,384</td>
<td>2</td>
<td>17</td>
<td>85.0</td>
<td>63.5</td>
</tr>
<tr>
<td>2</td>
<td>401</td>
<td>1,450</td>
<td>4</td>
<td>66</td>
<td>98.5</td>
<td>86.6</td>
</tr>
<tr>
<td>3</td>
<td>401</td>
<td>1,400</td>
<td>2</td>
<td>34</td>
<td>97.1</td>
<td>79.7</td>
</tr>
<tr>
<td>4</td>
<td>401</td>
<td>1,450</td>
<td>4</td>
<td>66</td>
<td>98.5</td>
<td>36.6</td>
</tr>
<tr>
<td>5</td>
<td>401</td>
<td>1,450</td>
<td>4</td>
<td>66</td>
<td>98.5</td>
<td>86.6</td>
</tr>
<tr>
<td>6</td>
<td>401</td>
<td>1,450</td>
<td>2</td>
<td>17</td>
<td>85.5</td>
<td>63.5</td>
</tr>
<tr>
<td>7</td>
<td>401</td>
<td>1,550</td>
<td>3</td>
<td>69</td>
<td>98.5</td>
<td>59.1</td>
</tr>
<tr>
<td>8</td>
<td>4,000</td>
<td>20,030</td>
<td>2</td>
<td>18</td>
<td>94.4</td>
<td>78.2</td>
</tr>
<tr>
<td>9</td>
<td>4,000</td>
<td>20,030</td>
<td>1</td>
<td>18</td>
<td>94.4</td>
<td>91.3</td>
</tr>
<tr>
<td>10</td>
<td>4,000</td>
<td>20,110</td>
<td>2</td>
<td>55</td>
<td>98.2</td>
<td>95.4</td>
</tr>
</tbody>
</table>

Notes: We report in the RF and ARVal columns the smallest value of \( \Gamma \), when a robust flow vanishes and an adaptive flow vanishes, respectively. The heuristically found adaptive flows are always close to the upper bound (Ubd), while a maximum flow (MF) is up to 40\% worse (percentages averaged over all nonvanishing choices of \( \Gamma \)).
 ainda

flow. It is a standard procedure to transform these problems into equivalent $s$-$t$-flows. In the motivating example of a pipeline system, we face a similar situation. Originally, there are multiple sources (and possibly more than one sink). A super source is introduced to phrase this problem as an $s$-$t$-flow. Of course, artificial arcs cannot be destroyed, and are hence rightly excluded from the failure sets. Note that this exclusion is in favor of the robust maximum flow approach, ameliorating its conservativeness. Still, the values of computed robust flows are significantly smaller than the adaptive flow values in this study.

7. Conclusions

This paper has presented a methodology to take network flow type decisions with respect to arc and node failures. Network flows display clean algebraic and combinatorial properties that allow for efficient algorithms. Still, in most practical applications data is uncertain, imprecise, or subject to changes. In particular, it stands to reason that some of the arcs or nodes in the network may fail. This causes significant changes for the optimization leading to problems that lose a lot of the advantageous properties of nominal network flows.

In this work, we have studied robust and adaptive versions of the maximum flow problem and the minimum cut problem with arc failures, and established structural properties and tractability results. We showed that the robust maximum flow problem is solvable in polynomial time, whereas the robust minimum cut problem and the adaptive versions are NP-hard. We also studied a different version of the adaptive model using a path-based formulation, where the flows are defined in paths, rather than arcs. We proposed an LO-based approximation method for the adaptive versions and demonstrated its computational capability to produce near-optimal solutions. We further extended the adaptive model to flows over time. We established that an adaptive maximum flow over time is not necessarily obtained by restricting our attention to the class of temporally repeated flows and the problem is weakly NP-hard, even when only one arc can fail.

Table 2. A tree-like instance with 118 nodes and 540 arcs, specifically designed for robust flows.

<table>
<thead>
<tr>
<th>$\Gamma$</th>
<th>Values of robust flows</th>
<th>Adjustable robust flow values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>461,431</td>
<td>476,123</td>
</tr>
<tr>
<td>2</td>
<td>340,370</td>
<td>3,941,182</td>
</tr>
<tr>
<td>3</td>
<td>228,796.621</td>
<td>319,512</td>
</tr>
<tr>
<td>4</td>
<td>110,159.517</td>
<td>247,073</td>
</tr>
<tr>
<td>5</td>
<td>32,937.7143</td>
<td>191,809</td>
</tr>
<tr>
<td>6</td>
<td>3,389.75</td>
<td>139,366</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>96,476</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>57,548</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>21,143</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Note. Still, adaptive flows get better values and allow for higher values of $\Gamma$ before they vanish.
We also explored a game-theoretic aspect of the adaptive model. More precisely, we characterized the adaptive model as a two-person zero-sum game in which the first player chooses a flow and the second player selects a cut. However, one might characterize the adaptive model from a different perspective: the first player chooses a flow and the second one selects a number of arcs to be eliminated. The two characterizations are equivalent if the first player chooses a flow in advance, but are different when the second player has to take his action first. The second characterization is also interesting because it corresponds to the network interdiction problem if the second player makes his decision first and corresponds to the adaptive maximum flow problem if the second player has to implement a flow up-front.

Acknowledgments

The authors thank the two reviewers and the associate editor for valuable suggestions and comments, which helped to improve the paper. Research of the second author was partially supported by the German Research Foundation [DFG-project SK58/7-1] and the Office of Naval Research [Grant N000141110056]. Research of the third author was partially supported by Marie Curie IO Fellowship ROSES [PIOF-GA-2009-254402].

References


**Dimitris Bertsimas** is the Boeing Professor of Operations Research and the Codirector of the Operations Research Center at the Massachusetts Institute of Technology. He is a member of the National Academy of Engineering and has received numerous research awards, including the Farkas Prize (2008), the Erlang Prize (1996), and the SIAM Prize in Optimization (1996). In the last decade his research has focused on robust optimization. The present paper is part of his ongoing work of revisiting classical optimization problems under uncertainty.

**Ebrahim Nasrabadi** is a postdoctoral associate at the Massachusetts Institute of Technology’s Sloan School of Management and Operations Research Center. He received the Certificate of Excellence from Berlin Mathematical School for his Ph.D. thesis. His primary areas of interest include dynamic optimization and decision making under uncertainty, and their applications in supply chain optimization, inventory and production systems, and revenue management problems.

**Sebastian Stiller** studied philosophy and mathematics at Friedrich-Alexander-Universität Erlangen, Germany, and Katholieke Universiteit Leuven, Belgium. The German Operations Research Society honored his thesis with its Ph.D. award. He also won a Marie Curie International Outgoing Fellowship of the European Commission. After a postdoctoral fellowship at the Massachusetts Institute of Technology’s Sloan School of Management, he returned to Technische Universität Berlin. His research interests range from robust optimization to network flows, scheduling, and game theory. His theoretical work and industry projects pertain to applications in traffic, transport, and logistics as well as finance and real-time systems.