Robust optimization with incremental recourse

Ebrahim Nasrabadi, James B. Orlin

Sloan School of Management and Operations Research Center, Bldg. E40-147, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
E-mail: {nasrabad,jorlin}@mit.edu

Abstract

In this paper, we consider an adaptive approach to address optimization problems with uncertain cost parameters. Here, the decision maker selects an initial decision, observes the realization of the uncertain cost parameters, and then is permitted to modify the initial decision. We treat the uncertainty using the framework of robust optimization in which uncertain parameters lie within a given set. The decision maker optimizes so as to develop the best cost guarantee in terms of the worst-case analysis. The recourse decision is "*incremental*"; that is, the decision maker is permitted to change the initial solution by a small fixed amount. We refer to the resulting problem as the *robust incremental* problem. We study robust incremental variants of several optimization problems. We show that the robust incremental counterpart of a linear program is itself a linear program if the uncertainty set is polyhedral. Hence, it is solvable in polynomial time. We establish the NP-hardness for robust incremental linear programming for the case of a discrete uncertainty set. We show that the robust incremental shortest path problem is NP-complete when costs are chosen from a polyhedral uncertainty set, even in the case that only one new arc may be added to the initial path. We also address the complexity of several special cases of the robust incremental shortest path problem and the robust incremental minimum spanning tree problem.

Keywords: robust optimization, incremental optimization, network optimization, complexity

1. Introduction

Researchers in the optimization community have developed a variety of approaches for addressing problems of optimization under uncertainty. In general, there are two main approaches to address uncertainty in optimization models: stochastic optimization and robust optimization. The former approach treats the uncertainty in the data as random variables, thus giving a rich set of modeling tools. These models typically lead to problems that are quite challenging to solve. It is also a practical challenge to determine probability distributions that accurately model the uncertainty. For more information on stochastic optimization, see [17, 8, 24]. In contrast, robust optimization models the uncertainty in a deterministic manner. It assumes that the uncertain parameters come from known sets. It seeks solutions with the best worst-case cost guarantee. We refer the reader to [3, 5] for a discussion of the theory and applications of robust optimization. In this paper, we consider a modeling framework of robustness that allows the decision maker to adjust the initial solution by a bounded amount after the uncertain data is realized. In what follows, we provide a mathematical description of our problem and then review the related literature.

Problem description. We consider optimization problems of the following form:

$$\min \quad \mathbf{c}^T \mathbf{x}$$
$$\text{s.t.} \quad \mathbf{x} \in \mathcal{S}, \tag{1}$$

where $\mathcal{S} \subseteq \mathbb{R}^n$ denotes the feasible region and $\mathbf{c} \in \mathbb{R}^n$ is the vector of cost parameters. For a fixed cost vector $\mathbf{c}$, we refer to the above problem as the *nominal* problem. Notice that the vector $\mathbf{c}$ is a column vector, and the superscript T denotes the transpose operation. For clarity of the presentation, we will denote all vectors by bold small letters and all matrices by bold capital letters.

We assume that the cost parameters are subject to uncertainty. We let $\mathcal{U} \subseteq \mathbb{R}^n$ be the *uncertainty set*, that is, the set of all possible cost vectors. If $\mathbf{x}$ is a feasible solution, the objective function value for $\mathbf{x}$ in the robust optimization problem is $\max\{\mathbf{c}^T\mathbf{x} : \mathbf{c} \in \mathcal{U}\}$, which corresponds to the worst-case cost for solution $\mathbf{x}$. The mathematics can be viewed in terms of an *adversary* who maliciously wants to increase the objective function value. The robust optimization problem is to find a minimax solution; that is, the feasible solution with the best worst-case cost guarantee.

We further assume that the decision maker is able to make an incremental change in the solution $\mathbf{x}$ after the uncertain cost parameters are revealed. We let $\mathcal{S}_\mathbf{x} := \{\mathbf{y} \in \mathcal{S} : F(\mathbf{x}, \mathbf{y}) \leq K\}$ be the set of all possible choices, where $F(\mathbf{x}, \mathbf{y})$ is a measure of the distance between solutions $\mathbf{x}$ and $\mathbf{y}$, and K is a given upper bound on the total distance permitted. We refer to the set $\mathcal{S}_\mathbf{x}$ as the *incremental set* and to the function F as the *incremental function*.

The goal is to find the best initial solution $\mathbf{x}$, assuming the worst-case cost scenario occurs and the decision maker is then allowed to transform the solution $\mathbf{x}$ into another solution $\mathbf{y}$ subject to the constraint that $F(x, y) \leq K$. This leads to the following *robust incremental* optimization problem

$$Z_{\text{RobInc}} := \min_{\mathbf{x} \in \mathcal{S}} \mathbf{d}^T\mathbf{x} + \max_{\mathbf{c} \in \mathcal{U}} \min_{\mathbf{y} \in \mathcal{S}_\mathbf{x}} \mathbf{c}^T\mathbf{y}. \tag{2}$$

Notice that the objective function includes a term $\mathbf{d}^T\mathbf{x}$, where $\mathbf{d}$ is the cost vector for the initial decision.

We illustrate our model with a shortest path problem under uncertainty. Suppose that a commuter needs to determine a route to be taken each day. Occasionally, roadwork causes certain routes to be unavailable or to have extra delays. In this case, the robust incremental shortest path problem would be to select an initial route so as to minimize the worst-case travel given (1) the possible route delays are from a specified set $\mathcal{U}$, and (2) the modified route can vary from the initial route by a limited "distance" K. Another possible example is slowing or stopping an outbreak of a disease by imposing selected quarantines or limits on travel. As the disease progresses, some interdictions may become very costly, and the solution needs to be adjusted. But in incremental optimization, adjustments to the original plan must be limited in scope.

Problem (2) includes several interesting problems as special cases. If $\mathbf{d} = 0$ and if $\mathcal{S}_\mathbf{x} = \{\mathbf{x}\}$, it reduces to the following *robust* problem:

$$Z_{\text{Rob}} := \min_{\mathbf{x} \in \mathcal{S}} \max_{\mathbf{c} \in \mathcal{U}} \mathbf{c}^T\mathbf{x}. \tag{3}$$

If $\mathbf{d} = 0$ and $\mathcal{S}_\mathbf{x} = \mathcal{S}$, then Problem (2) reduces to the following *adversarial* problem:

$$Z_{\text{Adv}} := \max_{\mathbf{c} \in \mathcal{U}} \min_{\mathbf{y} \in \mathcal{S}_\mathbf{x}} \mathbf{c}^T\mathbf{y}. \tag{4}$$

The inner minimization problem in Problem (2) corresponds to the following *incremental* problem:

$$Z_{\text{Inc}} := \min_{\mathbf{y} \in \mathcal{S}_\mathbf{x}} \mathbf{c}^T\mathbf{y}, \tag{5}$$

which addresses the case where $\mathbf{x}$ is fixed and the uncertain data is realized, and the goal is to make an improved subsequent decision. Notice that Problem (2), in general, does not include the incremental problem (5). However, for the case where $\mathcal{S} \subseteq \{0, 1\}^n$, Problem (5) becomes a special case of Problem (2) upon setting d_i to be a very large value ($d_i \geq nC$ would be enough where $C := \max\{c_i : i = 1, \ldots, n\}$) if $x_i = 0$ and $d_i = 0$, otherwise. Moreover, we require to set $\mathcal{U} = \{\mathbf{c}\}$. We then observe that the best initial solution for Problem (2) is $\mathbf{x}$. Since $\mathcal{U} = \{\mathbf{c}\}$, the problem reduces to the incremental problem.

Literature review. Robust optimization has been well studied in the literature. In the case that the nominal problem is convex and the uncertainty set is polyhedral or ellipsoidal, tractable reformulations are known (see, e.g., [2, 5] and the references therein). For 0–1 discrete optimization problems with uncertain cost parameters, Bertsiams and Sim [6] developed efficient algorithms for special cases in which the nominal problem is efficiently solvable. Goetzmann *et al.* [14] extended the results in [6] for integer programs with uncertain cost parameters and for integer programs with uncertainty in one or few constraints.

In robust optimization, the decision maker must determine all the decisions simultaneously, and in particular before the realization of the uncertainty. This assumption leads to a single-stage optimization problem. Ben-Tal *et al.* [3]

proposed a two-stage robust optimization modeling approach, the so-called *adjustable robust* optimization, in which the decision maker makes two sets of decisions: one set before the uncertainty being realized and one set subsequently. They showed that the adjustable robust counterpart of a linear program with right hand side uncertainty is, in general, NP-hard. Subsequently, a number of researchers have attempted to obtain approximations to the adjustable robust optimization problem. Significant results have been obtained in designing approximation algorithms for adjustable robust optimization problems. We refer the reader to [3, 5] and the references therein for a review of the literature on adjustable robust optimization. It is worthwhile pointing out that the robust incremental problem is a specific case of adjustable robust optimization and fits into the class of recoverable robustness. While previous research on adjustable optimization has mainly worked on problems with convex uncertainty sets, our main focus is on problems in which the uncertain costs reside within a discrete set.

In 2009, Liebchen *et al.* [18] introduced a new concept of robust adjustable optimization, so-called *recoverable robustness*, by allowing a limited recovery after the realization of uncertain parameters. They applied their modeling methodology to linear programming problems and provided an efficient algorithm in the case of right-hand side disturbances. Later, Büsing [9] specialized the concept of recoverable robustness to the shortest path problem and presented hardness results and approximation algorithms for different variants of the problem. We point out that the robust incremental problem fits into the class of recoverable robustness introduced by Liebchen *et al.* [18] and is quite similar to the recoverable robust shortest path problem studied by Büsing [9].

In robust optimization, the decision maker first chooses a solution, and then adversary selects the cost vector. In the adversarial problem, the order of moves is reversed: the adversary first secrets the cost vector, and then the decision maker selects a solution. Here, the adversary seeks a cost vector to maximize the optimal value of the solution chosen by the decision maker. This generalizes *the most vital variable problem*, which is to identify a variable whose cost increase leads to the largest increase in the optimal value (see, e.g., [27]). This includes the problem of finding most vital arcs in network optimization, such as in the maximum flow problem [28, 29, 22], the minimum spanning tree problem [19, 25] and the shortest path problem [1, 16, 21].

In incremental optimization, it is assumed that an initial solution is given, and the aim is to make an incremental change in the solution that will result in the greatest improvement in the objective function. The incremental counterpart of several network flow problems was studied by Şeref *et al.* [10]. They presented a polynomial algorithm for the incremental minimum spanning tree problem. They showed that the incremental minimum cost flow problem can be solved in polynomial time using Lagrangian relaxation. They also considered two versions of the incremental minimum shortest path problem, where increments are measured via arc inclusions and arc exclusions. They presented a polynomial time algorithm for the arc inclusion version and showed that the arc exclusion version is NP-complete.

Modeling assumptions and notation. The analysis of Problem (2) depends on the set $\mathcal{S}$, the uncertainty set $\mathcal{U}$, and the incremental set $\mathcal{S}_{\mathbf{x}}$ (or equivalently the incremental function F). We study linear and integer optimization problems under different types of uncertainty sets and incremental functions. To model the uncertainty set, we assume that the nominal costs are given by $\bar{c}_i, i = 1, \ldots, n$ and the cost of i-th cost coefficient can increase by at most $\hat{c}_i$. Thus, any possible realization of the i-th cost coefficient lies within the interval $[\bar{c}_i, \bar{c}_i + \hat{c}_i]$. Given a parameter $\Gamma > 0$, we consider the following two uncertainty sets:

$$\mathcal{U}_1 := \{\mathbf{c} = \bar{\mathbf{c}} + \boldsymbol{\delta} : \mathbf{0} \leq \boldsymbol{\delta} \leq \hat{\mathbf{c}}, \ \sum_{i=1}^{n} \delta_i \leq \Gamma\}, \tag{6}$$

$$\mathcal{U}_2 := \{\mathbf{c} = \bar{\mathbf{c}} + \boldsymbol{\delta} \cdot \hat{\mathbf{c}} : \boldsymbol{\delta} \in \{0, 1\}^n, \sum_{i=1}^{n} \delta_i \leq \Gamma\}, \tag{7}$$

where $\boldsymbol{\delta} \cdot \hat{\mathbf{c}}$ is the doc product of the vectors $\boldsymbol{\delta}$ and $\hat{\mathbf{c}}$. The uncertainty set $\mathcal{U}_1$ models the case where the total change permitted in the cost coefficients is limited by Γ, whereas in the uncertainty set $\mathcal{U}_1$, the cost of at most Γ coefficients is allowed to increase from their nominal costs (here Γ is assumed to be integer). The extreme points of the set $\mathcal{U}_1$ are the cost vectors in $\mathcal{U}_2$. Hence, the two sets lead to equivalent robust optimization problems. But this is not the case for the adversarial problem and the robust incremental problem as the two sets lead to different problems.

Because robust incremental problems involve three stages of decisions, it is unlikely that they will be in the class NP (assuming that the polynomial time hierarchy does not collapse.) For example, consider the special case of the

Table 1: Complexity of different problems

Problem/Unc/Dist	Z_{Rob}	Z_{Inc}	Z_{Adv}	Z_{RobInc}
LP/$\mathcal{U}_1$/L_1	P [4, 7]	P [10]	**P**	**P**
LP/$\mathcal{U}_2$/L_1	P [4, 7]	P [10]	**NPC**	**NPH**
SP/$\mathcal{U}_1$/Incl	P [6]	P [10]	**P**	**NPC**
SP/$\mathcal{U}_1$/Excl	P [6]	NPC [10]	NPC [10]	NPH [10]
SP/$\mathcal{U}_1$/Sym	P [6]	**NPC**	**NPH**	**NPH**
SP/$\mathcal{U}_2$/Incl	P [6]	P [10]	NPC [1]	**NPH** [1, 9]
SP/$\mathcal{U}_2$/Excl	P [6]	NPC [10]	NPC [1]	NPH [1]
SP/$\mathcal{U}_2$/Sym	P [6]	**NPC**	NPC [1]	**NPH** [1]
MST/$\mathcal{U}_1$/Incl	P [6]	P [10]	**P**	– –
MST/$\mathcal{U}_2$/Incl	P [6]	P [10]	NPC [19]	NPH [19]

robust incremental problem (2) in which S denotes the feasible region of a combinatorial optimization problem, and where $\mathcal{U} = \mathcal{U}_1$, and $S_x = \{y : \sum_{i=1}^n |x_i - y_i| \le K\}$ for some specified K. Suppose that the nominal optimization problem is solvable in polynomial time. In this case, the decision variant of the third stage problem, i.e., the incremental problem, is in the class P. (The decision variant is a yes-no variant of the optimization problem.) The decision variant of the second stage decision, i.e., the adversarial problem, is in the class NP, and the decision variant of the robust incremental problem is in the class Σ_2^p, which is one level higher than NP in the polynomial time hierarchy (see, e.g., [20, 26]). If the decision variant of the third stage problem, i.e., the incremental problem, is in NP, then the adversarial problem is in the class Σ_2^p. And the decision variant of the robust incremental problem is in the class Σ_3^p, which is one level higher than Σ_2^p on the polynomial time hierarchy.

In this paper, we address problems in which the nominal problem is in the class P. In particular, we consider robust incremental variants of the following three types of problems: (1) linear programming problems, (2) shortest path problems, and (3) minimum spanning tree problems. In cases in which the adversarial problem is NP-hard, it remains an open question as to whether the decision variant of the robust incremental problem Σ_2^p-complete.

We consider three types of incremental functions. For linear programming, we restrict attention to the L_1–distance function, that is, $F(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$. For the shortest path problem, we consider two additional incremental functions. In measuring the distance of a path P from a given path P^0, we consider the arc inclusion function $|P \setminus P^0|$ and the arc exclusion function $|P^0 \setminus P|$. We note that the L_1–distance in this case is equivalent to the arc symmetric-difference $|P^0 \oplus P|$. For the minimum spanning tree problem, the complexity for all three metrics is equivalent, and we measure the distance of a tree T from a given tree T^0 by the arc inclusion function $|T \setminus T^0|$.

Our contribution. Our primary contributions are as follows:

1. We show that the robust incremental counterpart of a linear program with respect to uncertainty set $\mathcal{U}_1$ is transformable into a linear program and consequently solvable in polynomial time. In the case that the uncertainty set is $\mathcal{U}_2$, we show that robust incremental linear programming is NP-hard. In particular, we show that the adversarial minimum cost flow problem is NP-hard with respect to the discrete uncertainty set $\mathcal{U}_2$.

2. We show that the robust incremental shortest path problem is NP-hard even in the special case that the uncertainty set is $\mathcal{U}_1$ and only one new arc may be added to the initial path.

3. We prove that the adversarial shortest path problem is solvable in polynomial time with respect to the uncertainty set $\mathcal{U}_1$ and the arc inclusion function, while the robust incremental problem is NP-hard. We also show that the incremental shortest path problem is NP-hard with respect to the arc symmetric-difference function.

4. We also consider an adversarial variant of the minimum spanning tree problem. We show that the adversarial minimum spanning tree problem is solvable in polynomial time when the cost parameters lie in the uncertainty set $\mathcal{U}_1$.

4

We refer to Table 1 for a summary of our contributions as well as a summary of known results from the literature. The input for the problem is given as a triple. The first term of the triple is LP, SP, or MST indicating linear programs, shortest paths or minimum spanning trees. The second term of the triple is $\mathcal{U}_1$ or $\mathcal{U}_2$, indicating the type of uncertainty set. The third term of the triple is the type of distance metric. It is either L_1 or it is an abbreviation for set inclusion, set exclusion, or set symmetric difference. We indicate the complexity of the decision version of the problem as P (polynomially solvable), NPC (NP-complete) or NPH (NP-hard). If the complexity is stated with a bold font, it refers to results established in this paper. For SP/$\mathcal{U}_2$/Incl, Büsing [9] shows that the problem is NP-hard for constant $K \geq 1$. Here, we show that the problem is NP-hard, even for $\Gamma = K = 1$. For SP/$\mathcal{U}_1$/Incl, we prove that the problem is NP-hard and inapproximable within within a factor of 2, even for $\Gamma = K = 1$ Of the 28 different problems [1] summarized in Table 1, only one of the problems remains open.

2. Robust-Incremental Linear Programming

In this section, we consider the robust incremental optimization model for linear programming. More precisely, we assume that the feasible region $\mathcal{S}$ is given by non-negativity constraints as well as a number of linear equalities, that is,

$$\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{Ax} = \mathbf{b}\}, \tag{8}$$

where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{b}$ is an m-vector of the right-hand side parameters.

We first consider the uncertainty set $\mathcal{U}_1$, where the total change in the cost parameters is bounded by Γ. We further assume that the total change permitted in the initial solution is bounded by K. The incremental set is given by

$$\mathcal{S}_{\mathbf{x}} := \{\mathbf{y} \in \mathcal{S} : \sum_{i=1}^{n} |x_i - y_i| \leq K\}. \tag{9}$$

We show that the resulting robust incremental optimization problem may be represented as a linear programming problem, and hence can be solved efficiently. With respect to a solution $\mathbf{x} \in \mathcal{S}$ and a cost vector $\mathbf{c} \in \mathcal{U}$, we define $Z_{\text{Inc}}(\mathbf{x}, \mathbf{c})$ to be the optimal value of the incremental problem, that is,

$$
\begin{aligned}
Z_{\text{Inc}}(\mathbf{x}, \mathbf{c}) := \min \quad & \mathbf{c}^T \mathbf{y} \\
\text{s.t.} \quad & Ay = \mathbf{b}, \\
& \sum_{i=1}^{n} |x_i - y_i| \leq K, \\
& \mathbf{y} \geq \mathbf{0}.
\end{aligned}
\tag{10}
$$

We introduce two nonnegative variables z_i^+, z_i^-, and set $y_i - x_i = z_i^+ - z_i^-$ for $i = 1, \ldots, n$. The variables z_i^+ and z_i^- correspond to the increase or decrease in x_i, respectively. There is always an optimal solution in which either $z_i^+ = 0$ or $z_i^- = 0$, in which case $|z_i^+ - z_i^-| = z_i^+ + z_i^-$. Hence, $Z_{\text{Inc}}(\mathbf{x}, \mathbf{c})$ is equivalent to the following linear program:

$$
\begin{aligned}
Z_{\text{Inc}}(\mathbf{x}, \mathbf{c}) = \min \quad & \mathbf{c}^T \mathbf{x} + \mathbf{c}^T (\mathbf{z}^+ - \mathbf{z}^-) \\
\text{s.t.} \quad & \mathbf{A}(\mathbf{z}^+ - \mathbf{z}^-) = \mathbf{0}, \\
& \sum_{i=1}^{n} (z_i^+ + z_i^-) \leq K, \\
& \mathbf{z}^- \leq \mathbf{x}, \\
& \mathbf{z}^+, \mathbf{z}^- \geq \mathbf{0}.
\end{aligned}
\tag{11}
$$

---

[1] Notice that some problems in the table are equivalent. For example, LP/$\mathcal{U}_1$/L_1 is equivalent to LP/$\mathcal{U}_2$/L_1 for the robust problem. In total, there are 28 different problems in Table 1.

The dual of the above problem is the following linear program:

$$
\begin{aligned}
Z_{\text{Inc}}(\mathbf{x}, \mathbf{c}) = \max \quad & \mathbf{c}^T \mathbf{x} - \alpha K - \mathbf{x}^T \mathbf{v} \\
\text{s.t.} \quad & \mathbf{w}^T \mathbf{A} - \alpha \mathbf{1} \leq \mathbf{c}, \\
& -\mathbf{w}^T \mathbf{A} - \alpha \mathbf{1} - \mathbf{v} \leq -\mathbf{c}, \\
& \mathbf{v} \geq \mathbf{0}, \\
& \alpha \geq 0,
\end{aligned}
\tag{12}
$$

where $\mathbf{w}$ is an m-vector, $\mathbf{v}$ is an n-vector, and $\mathbf{1}$ is an m-vector each of whose entries is one. Then $\mathbf{w}$ is the vector of dual variables corresponding to the first set of constraints, $-\alpha$ is a dual variable corresponding to the second constraint, and $\mathbf{v}$ is the vector of dual variables corresponding to the third set of constraints in Problem (11).

We now define $Z_{\text{Adv}}(\mathbf{x}) := \max_{\mathbf{c} \in \mathcal{U}_1} Z_{\text{Inc}}(\mathbf{x}, \mathbf{c})$ to be the optimal value of the adversarial problem with respect to a given solution $\mathbf{x}$. Following the above discussion, $Z_{\text{Adv}}(\mathbf{x})$ can be expressed as follows:

$$
\begin{aligned}
Z_{\text{Adv}}(\mathbf{x}) = \max \quad & \bar{\mathbf{c}}^T \mathbf{x} + \delta^T \mathbf{x} - \alpha K - \mathbf{x}^T \mathbf{v} \\
\text{s.t.} \quad & \mathbf{w}^T \mathbf{A} - \alpha \mathbf{1} - \delta \leq \bar{\mathbf{c}}, \\
& -\mathbf{w}^T \mathbf{A} - \alpha \mathbf{1} - \mathbf{v} + \delta \leq -\bar{\mathbf{c}}, \\
& \sum_{i=1}^{n} \delta_i \leq \Gamma, \\
& 0 \leq \delta \leq \hat{\mathbf{c}}, \\
& \mathbf{v} \geq \mathbf{0}, \\
& \alpha \geq 0.
\end{aligned}
\tag{13}
$$

We once again take the dual and obtain:

$$
\begin{aligned}
Z_{\text{Adv}}(\mathbf{x}) = \min \quad & \bar{\mathbf{c}}^T (\mathbf{x} + \mathbf{z}^+ - \mathbf{z}^-) + \beta \Gamma + \hat{\mathbf{c}}^T \mathbf{q} \\
\text{s.t.} \quad & \mathbf{A}(\mathbf{z}^+ - \mathbf{z}^-) = 0, \\
& \sum_{i=1}^{n} (z_i^+ + z_i^-) \leq K, \\
& -\mathbf{z}^+ + \mathbf{z}^- + \beta \mathbf{1} + \mathbf{q} \geq \mathbf{x}, \\
& \mathbf{z}^- \leq \mathbf{x}, \\
& \mathbf{z}^+, \mathbf{z}^-, \mathbf{q} \geq \mathbf{0}, \\
& \beta \geq 0.
\end{aligned}
\tag{14}
$$

The goal of Problem (2) is find a solution $\mathbf{x} \in \mathcal{S}$ with minimum value $Z_{\text{Adv}}(\mathbf{x})$. This establishes the following theorem.

Theorem 1. *Suppose that the sets $\mathcal{S}$, $\mathcal{U}$ and $\mathcal{S}_{\mathbf{x}}$ are given by (8), (6), and (9), respectively. Then, the robust incre-*

mental optimization Problem (2) *may be formulated as the following linear programming problem:*

$$\min \quad \mathbf{d}^T\mathbf{x} + \bar{\mathbf{c}}^T(\mathbf{x} + \mathbf{z}^+ - \mathbf{z}^-) + \beta\Gamma + \hat{\mathbf{c}}^T\mathbf{q}$$

$$s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{b},$$

$$\mathbf{A}(\mathbf{z}^+ - \mathbf{z}^-) = 0,$$

$$\sum_{i=1}^{n}(z_i^+ + z_i^-) \leq K,$$

$$-\mathbf{z}^+ + \mathbf{z}^- + \beta\mathbf{1} + \mathbf{q} \geq \mathbf{x},$$

$$\mathbf{z}^- \leq \mathbf{x},$$

$$\mathbf{z}^+, \mathbf{z}^-, \mathbf{q} \geq \mathbf{0},$$

$$\mathbf{x} \geq \mathbf{0},$$

$$\beta \geq 0.$$

(15)

Therefore, Problem (2) *can be solved in polynomial time.*

We next turn our attention to the uncertainty set $\mathcal{U}_2$ and show that the robust incremental counterpart of a linear program in NP-hard. To this end, we considet the minimum cost flow problem. In this problem, we are given a directed graph $G = (N, A)$ with *node set N* and *arc set $A \subseteq N \times N$*. Each arc $(i, j) \in A$ has an associated *capacity u_{ij}* and an associated *cost c_{ij}*. The supply/demand of node i is b_i. We assume that $\sum_{i \in N} b_i = 0$.

A *flow* is a vector $\mathbf{x} \in \mathbb{R}_+^{|A|}$ that assigns a nonnegative value x_{ij} to arc (i, j). We refer to x_{ij} as the *flow on arc* (i, j). We require that a flow $\mathbf{x}$ obeys the *capacity constraints $x_{ij} \leq u_{ij}$, $(i, j) \in A$* and satisfies the following *flow conservation constraints*:

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \qquad\qquad \forall i \in N.$$

The *cost* of a flow $\mathbf{x}$ is given by $\sum_{(i,j)\in A} c_{ij}x_{ij}$. The *minimum cost flow problem* is to find a feasible flow of minimum cost. This problem can be stated as follows:

$$\min \quad \mathbf{c}^T\mathbf{x}$$

$$s.t. \quad \mathbf{N}\mathbf{x} = \mathbf{b},$$

$$0 \leq \mathbf{x} \leq \mathbf{u},$$

(16)

where $\mathbf{N}$ is the node-arc adjacency matrix of the graph G, $\mathbf{u}$ is the vector of arc capacities, and $\mathbf{b}$ is the vector of supplies or demands.

We suppose that arc costs are uncertain. The cost of arc (i, j) is in the interval $[\bar{c}_{ij}, \bar{c}_{ij} + \hat{c}_{ij}]$. We assume that the total change permitted in the incremental stage is bounded by K. Given a flow $\mathbf{x}$, the *adversarial* minimum cost flow problem is to find a cost vector $\mathbf{c} \in \mathcal{U}_1$ that maximizes the optimal value $Z_{\text{Inc}}(\mathbf{x}, \mathbf{c})$ of the incremental problem.

In the case that the uncertainty set is $\mathcal{U}_1$, the results of the previous section show that the robust incremental min cost flow problem can be formulated as a linear programming problem, and is thus solvable in polynomial time.

We next show that the adversarial minimum cost flow problem is NP-hard under the uncertainty set $\mathcal{U}_2$.

The adversarial problem is to increase the costs of at most Γ arcs such as to maximize the minimum cost flow. This problem can be formulated as follows:

$$\max_{\delta \in \Theta} \quad \min \quad \sum_{(i,j)\in A} (\bar{c}_{ij}x_{ij} + \delta_{ij}\hat{c}_{ij}x_{ij})$$

$$s.t. \quad \mathbf{N}\mathbf{x} = \mathbf{b},$$

$$0 \leq \mathbf{x} \leq \mathbf{u},$$

(17)

where $\Theta := \{\boldsymbol{\delta} \in \{0, 1\}^{|A|} : \sum_{(i,j)\in A} \delta_{ij} \leq \Gamma\}$.

We show that this problem is NP-hard by a transformation from the *network interdiction* problem, which is defined as follows: reduce the value of a maximum flow in a given network as much as possible by removing Γ arcs. Wood [29] shows that this problem is NP-hard by a transformation from the clique problem.

Theorem 2. *Under the uncertainty set $\mathcal{U}_2$, the adversarial minimum cost flow problem is NP-hard.*

Proof. Consider the following decision version of the network interdiction problem. We are given a capacitated network $G = (N, A)$ with a source $s \in N$ and sink $t \in N$, an integer Γ, and an integer k. The problem is to determine whether there exists a set of Γ arcs so that after removing these arcs the maximum flow value is at most $k - 1$. We now construct an instance of the adversarial minimum cost flow problem as follows. Suppose that we want to ship k units of flow from s to t in the given network. We let $b_s = k$, $b_t = -k$, and $b_i = 0$ for all other nodes. Moreover, assume that the nominal costs $\bar{c}_{ij}$ are all zero and an adversary can increase the costs of at most Γ arcs by one, that is, $\bar{c}_{ij} = 0$ and $\hat{c}_{ij} = 1$ for all arcs $(i, j) \in A$.

We next show that the network interdiction flow problem is a "Yes"-instance if and only if the optimal value of the corresponding adversarial minimum cost flow problem is strictly positive. Suppose that the network interdiction flow problem is a "Yes"-instance. Thus, there are Γ arcs so that one cannot ship k units of flow from s to t without sending flow on at least one of these arcs. This implies that if the adversary increases the cost of these Γ arcs by one, then the cost for sending k units of flow will be strictly positive. Conversely, suppose that the optimal value of the adversarial minimum cost flow problem is strictly positive. Let S denote the set of Γ arcs whose cost was increased from 0 to 1. Then any flow of k units from s to t must use at least one of these arcs. This completes the proof of the theorem. $\square$

The above theorem implies that under the uncertainty set $\mathcal{U}_2$, robust incremental network flows (and thus robust incremental linear programming) is NP-hard. It is an open question whether the decision variant of robust incremental network flows is in the class NP. It is also open as to whether the problem is Σ_2^p-complete.

3. Robust-Incremental Shortest Path Problem

Here, we study the shortest path problem from a robust incremental viewpoint. Let $G = (N, A)$ be a directed graph with a *source* $s \in N$ and a *sink* $t \in N$, and let c_{ij} denote the cost (length) of arc $(i, j) \in A$. In the *shortest path problem*, we seek a path of least cost from s to t.

We suppose that for each arc (i, j), the cost c_{ij} is uncertain and can vary within the interval $[\bar{c}_{ij}, \bar{c}_{ij} + \hat{c}_{ij}]$. If the decision maker chooses a path P^0 in the first stage and observes the realization $\mathbf{c}$ of the cost vector, she is allowed to build a new path P in the second stage whose distance from P^0 is not more than K for some specified integer K. We measure the distance of a path P from the path P^0 via the three different incremental functions: $|P \setminus P^0|$, $|P^0 \setminus P|$, and $|P \oplus P^0|$.

If $K \geq 2n - 2$ and if the uncertainty set is $\mathcal{U}_2$, the *adversarial* shortest path problem is the problem of maximizing the length of the least cost path from s to t by increasing the cost of at most Γ arcs.

By setting $\hat{c}_{ij}$ to be very large, the adversarial shortest path problem reduces to the problem of determining the Γ most vital arcs, i.e., those Γ arcs whose removal results in the greatest increase in the length of the shortest path from s to t. The latter problem is known to be NP-hard and inapproximable by a constant factor better than 2 [see 1]. This implies that the robust incremental shortest path problem is NP-hard with respect to all three incremental functions. In the rest of this section, we examine the robust incremental shortest path problem with respect the uncertainty set $\mathcal{U}_1$, unless mentioned otherwise, and under the incremental functions $|P \setminus P^0|$ and $|P^0 \oplus P|$.

3.1. The incremental function $|P \setminus P^0|$

Here, we assume that one can build a new path by adding at most K new arcs to the path in the incremental stage. For this case, Şeref *et al.* [10] present a polynomial time algorithm for solving the incremental shortest path problem. We show that the adversarial problem is also solvable in polynomial time. And we show that the robust incremental problem is NP-hard, even for the special case where $\Gamma = K = 1$.

Let $Z_{\text{Inc}}(P^0, \mathbf{c})$ be the optimal value of the incremental optimization problem on $G = (N, A)$ with respect to a given path P^0 and a vector $\mathbf{c}$ of costs. We next transform the incremental optimization problem on G into a shortest path problem on a time expanded network $G^* = (N^*, A^*)$. We create the time expanded network G^* as follows. For each node $i \in N$ there are $K + 1$ copies of the node in N^*. The copies are denoted as i_k, for $k = 0, \ldots, K$.

There are three subsets of arcs. For every arc $(i, j) \in P^0$, there are arcs (i_k, j_k), for $k = 0, \ldots, K$. For every arc $(i, j) \in A \setminus P^0$, there are arcs (i_k, j_{k+1}), for $k = 0, \ldots, K - 1$. Finally, for every node $i \in N$, there are arcs (i_k, i_{k+1}), for
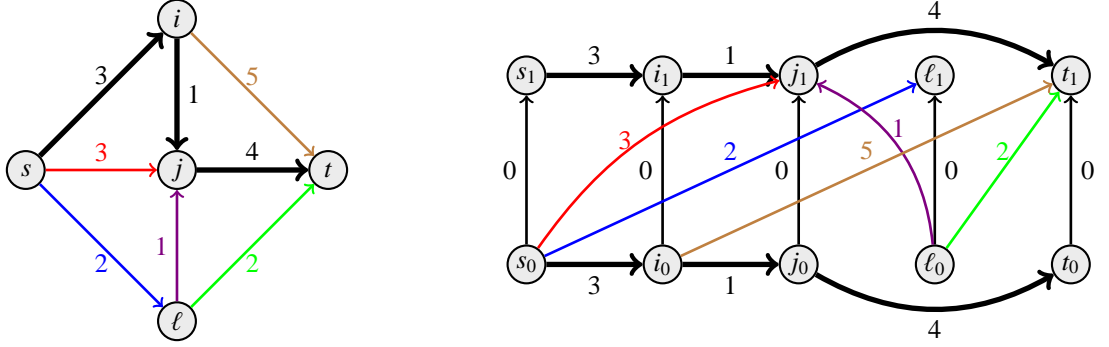
Figure 1: On the left hand side, a network G is shown. Let $P^0 = s, i, j, t$ and let $K = 1$. On the right hand side, the corresponding time expanded network G^* is depicted. The number on the arc indicates the cost.

$k = 0, \ldots, K - 1$. Arcs in the first two sets have the same cost as in G. Arcs in the third set have a cost of 0. The construction of the time expanded network is illustrated in Figure 1 for a simple network.

We see that the transformation is valid as follows. A feasible path P in the original network with $|P \setminus P^0| \leq K$ will induce a path Q from s_0 to t_K in the time expanded network so that the cost of P and Q are the same. Similarly, a feasible path Q from s_0 to t_K in the time expanded network induces a path P in G whose cost is the same as that of Q, and such that $|P \setminus P^0| \leq K$.

We have expressed the incremental optimization problem as a shortest path problem on the time expanded network. The latter problem is formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{k=0}^{K} \sum_{(i,j) \in P^0} c_{ij} x_{ij}^k + \sum_{k=0}^{K} \sum_{(i,j) \in A \setminus P^0} c_{ij} x_{ij}^k \\
\text{s.t.} \quad & x_{ii}^k + \sum_{j:(i,j) \in P^0} x_{ij}^k + \sum_{j:(i,j) \in A \setminus P^0} x_{ij}^k \\
& - x_{ii}^{k-1} - \sum_{j:(j,i) \in P^0} x_{ji}^k - \sum_{j:(j,i) \in A \setminus P^0} x_{ji}^{k-1} = \begin{cases} 1 & \text{if } i = s, k = 0 \\ -1 & \text{if } i = t, k = K \quad \forall i \in N, \ k = 0, \ldots, K, \\ 0 & \text{otherwise} \end{cases} \\
& x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in P^0, \ k = 0, \ldots, K, \\
& x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A \setminus P^0, \ k = 0, \ldots, K - 1, \\
& x_{ii}^k \in \{0,1\} \qquad \forall i \in N, \ k = 0, \ldots, K - 1, \\
& x_{ij}^K = 0 \qquad \forall (i,j) \in A \setminus P^0, \\
& x_{ii}^K = 0 \qquad \forall i \in N.
\end{aligned} \tag{18}
$$

In this problem, there are three subsets of decision variables corresponding to three subsets of arcs in the time expanded network. For every $(i,j) \in P^0$ and $k = 0, \ldots, K$, the decision variable x_{ij}^k corresponds to the arc from node i_k to node j_k. For every $(i,j) \in A \setminus P^0$ and $k = 0, \ldots, K - 1$, the decision variable x_{ij}^k corresponds to the arc from node i_k to node j_{k+1}. For every $i \in N$ and $k = 0, \ldots, K - 1$, the decision variable x_{ii}^k corresponds to the arc from node i_k to node i_{k+1}. There is a one-to-one correspondence between feasible solutions of Problem (18) and the paths from node s_0 to node t_K. We have included dummy variables x_{ij}^K for $(i,j) \in A \setminus P^0$ and x_{ii}^K for $i \in N$ to simplify the formulation, and then we let them to be zero.

We can relax the binary variables in Problem (18). By considering the dual problem, we obtain the following

formulation for the incremental shortest path problem:

$$Z_{\text{Inc}}(P^0, \mathbf{c}) = \max \quad \pi_s^0 - \pi_t^K$$
$$\text{s.t.} \quad \pi_i^k - \pi_j^k \le c_{ij} \quad \forall (i, j) \in P^0, \ k = 0, \ldots, K,$$
$$\pi_i^k - \pi_j^{k+1} \le c_{ij} \quad \forall (i, j) \in A \setminus P^0, \ k = 0, \ldots, K - 1, \tag{19}$$
$$\pi_i^k - \pi_i^{k+1} \le 0 \quad \forall i \in N, \ k = 0, \ldots, K - 1.$$

The adversarial problem is to find a cost vector $\mathbf{c} \in \mathcal{U}_1$ to maximize $Z_{\text{Inc}}(P^0, \mathbf{c})$. This implies that the adversarial problem turns into a linear program as follows:

$$Z_{\text{Adv}}(P^0) = \max \quad \pi_s^0 - \pi_t^K$$
$$\text{s.t.} \quad \pi_i^k - \pi_j^k - \delta_{ij} \le \bar{c}_{ij} \quad \forall (i, j) \in P^0, \ k = 0, \ldots, K,$$
$$\pi_i^k - \pi_j^{k+1} - \delta_{ij} \le \bar{c}_{ij} \quad \forall (i, j) \in A \setminus P^0, \ k = 0, \ldots, K - 1, \tag{20}$$
$$\pi_i^k - \pi_i^{k+1} \le 0 \quad \forall i \in N, \ k = 0, \ldots, K - 1,$$
$$0 \le \delta_{ij} \le \hat{c}_{ij} \quad \forall (i, j) \in A.$$

Notice that this problem has $Kn + m$ variables and at most $(K + 2)m + nK$ constraints. We assume without loss of generality that $K \le n - 1$ since otherwise the adversarial shortest path problem reduces to a nominal problem. This leads to the following result.

Theorem 3. *The adversarial shortest path problem under the uncertainty set $\mathcal{U}_1$ can be formulated as a linear program, and is solvable in polynomial time.*

Bertsimas and Sim [6] showed that the robust shortest path problem is solvable in polynomial time. We next prove that decision variant of the robust incremental shortest path problem is NP-complete. The transformation is from the *2-disjoint-paths problem*, which is as follows. Given given a directed graph $G = (N, A)$ and distinct nodes s_1, s_2, t_1, t_2, are there two disjoint paths P_1 and P_2 such that P_1 is from s_1 to t_1 and P_2 is from s_2 to t_2? This problem is shown to be NP-complete by Fortune *et al.* [13].

Theorem 4. *It is NP-hard to approximate the robust incremental shortest path problem within a factor of 2 if the uncertainty set is $\mathcal{U}_1$ and the distance metric is with respect to inclusion.*

Proof. Given an instance of the 2-disjoint-paths problem, we construct a graph $G' = (N', A')$ as follows. We create four nodes s_1', s_2', t_1', t_2', and arcs (s_1', s_1), (s_1', t_1'), (t_1, t_1'), (s_2', s_2), (s_2', t_2'), and (t_2, t_2'). We furthermore link node t_1' to node s_2' with two parallel arcs. The construction of graph G' is illustrated in Figure 2. Suppose that the costs d_{ij} for the initial path and the nominal arc costs $\bar{c}_{ij}$ are all zero, and suppose that an adversary can increase the cost of each arc by one. Furthermore, let $\Gamma = K = 1$.

We prove the following: (i) if the 2-disjoint-paths problem is a "yes" instance, then there exists a path P^0 from node s_1' to node t_2' in graph G' with $Z_{\text{Adv}}(P^0) = 0.5$, where $Z_{\text{Adv}}(P^0)$ is the optimal value of the adversarial optimization problem with respect to P^0; and (ii) if the 2-disjoint-paths problem is a "no" instance, then $Z_{\text{Adv}}(P^0) = 1$ for all paths P^0 from node s_1' to node t_2'.

We first suppose that there are two node-disjoint paths P_1 and P_2 from s_1 to t_1 and from s_2 to t_2, respectively. Let path P^0 be defined as follows:

$$P^0 := (s_1', s_1), P_1, (t_1, t_1'), (t_1', s_2'), (s_2', s_2), P_2, (t_2, t_s').$$

If the adversary were to modify the cost of one arc only, then the best incremental path would have a cost of 0. (This is easily seen by enumeration). Thus, the adversary needs to modify the costs of at least two arcs. An optimal choice for the adversary is to increase the costs of an arc of P_1 (or P_2) and the cost of the arc (t_1', s_2') used in P^0, each with 0.5. Then an optimal incremental path would be to replace the arc (t_1', s_2') used in P^0 by the other parallel arc (t_1', s_2'), which gives a cost of 0.5.
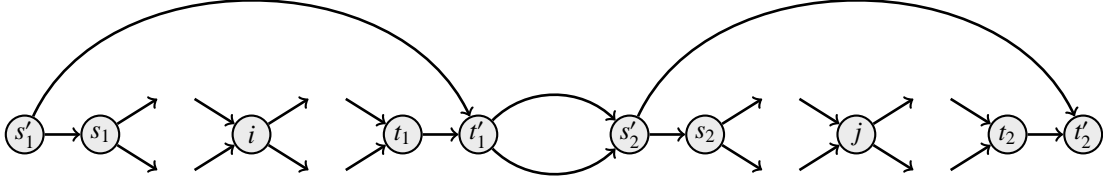
Figure 2: Construction of G' from G. The graph G' is constructed from G by introducing fours additional nodes s_1', s_2', t_1', t_2' and additional arcs $(s_1', s_1), (s_2', t_2'), (s_1', s_1), (t_2, t_2')$ and two parallel arcs from t_1' to s_2'.

We now assume that the 2-disjoint-paths problem is a "no" instance and prove that $Z_{\text{Adv}}(P^0) = 1$ for every path P^0 from node s_1' to node t_2'. Consider an arbitrary path P^0. We consider first the case in which P^0 passes through node t_1' (as well as node s_2'). In this case, it must contain arc (s_1', t_1') or arc (s_2', t_2'). We assume without loss of generality that it is arc (s_1', t_1'). In this case, the adversary increases the cost of arc (s_1', t_1') by 1. The incremental path must include this arc, and so has a cost of 1.

Next, we assume that the path P^0 does not pass through node t_1'. In this case, path P^0 must contain the arc (t_2, t_2'). In this case, the adversary increases the cost of arc (t_2, t_2') by 1. Every path P from node s_1' to node t_2' with $|P \setminus P^0| \leq 1$ must contain this arc as well, which implies that $Z_{\text{Adv}}(P^0) = 1$. This establishes the proof of the theorem. $\qquad\square$

We have already mentioned that the robust incremental shortest path problem is NP-hard under the uncertainty set $\mathcal{U}_2$ since it includes as special case the problem of finding the K mostvital arcs. However, for a fixed K, and in particular $K = 1$, the latter problem can be solved in polynomial time. The next result shows that the robust incremental shortest path problem is NP-hard with respect to the uncertainty set $\mathcal{U}_2$, even for $\Gamma = K = 1$.

Theorem 5. *The robust incremental shortest path problem is NP-hard under the uncertainty set $\mathcal{U}_2$.*

Proof. We use a similar reduction as in the proof of Theorem 5 from the 2-disjoint-paths problem. Consider an instance of the 2-disjoint-paths problem given by a directed graph $G = (N, A)$ and distinct nodes s_1, s_2, t_1, t_2. In addition, let G' be the constructed graph as in the proof of Theorem 5.

We show that the 2-disjoint-paths problem is a "yes" instance if and only if there exists a path P^0 from s_1' to t_2' with $Z_{\text{Adv}}(P^0) = 0$. We first suppose that there are two node-disjoint paths P_1 and P_2 from s_1 to t_1 and from s_2 to t_2, respectively. We set $P^0 = (s_1', s_1), P_1, (t_1, t_1'), (t_1', s_2'), (s_2', s_2), P_2, (t_2, t_s')$. If the cost of an arc which is not in the path P^0 is increased, then P^0 is still a path from s_1' to t_2' of cost zero. So we suppose that the cost of an arc in P^0 in increased by 1. If this arc is before node t_1' (after node s_2'), then we add arc (s_1', t_1') (arc (s_2', t_2')). In the case that the cost of arc (t_1', s_2') is increased, the other parallel arc can be replaced. Thus, in any case we can add one arc and have a new path from s_1' to t_2' of cost zero.

We next proceed to prove the reverse direction. Assume that there is path P^0 so that whenever the cost of one arc is increased, we can add at most one arc and have a path from s_1' to t_2' of cost zero. Such a path must not include either (s_1', t_1') or (s_2', t_2') since otherwise when the cost of one of these two arcs increases, we cannot build up any path of cost zero by adding one arc. This implies that there is a path from s_1 to t_1 and a path from s_2 to t_2. This completes the proof of the theorem. $\qquad\square$

3.2. The incremental function $|P \oplus P^0|$

While the incremental shortest path problem with arc inclusion can be solved efficiently, the arc exclusion version is NP-complete [see 10]. We next prove that the incremental shortest path problem is also NP-complete under the symmetric difference variant in which one is allowed to add or remove at most K arcs from the given path.

Theorem 6. *Under the uncertainty set $\mathcal{U}_1$, the increment shortest path problem is NP-complete.*
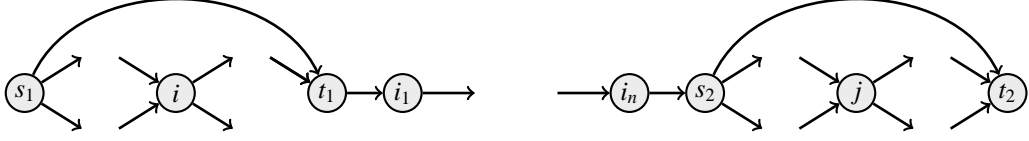
11

Figure 3: The graph G' is constructed from G by adding two new arcs (s_1, t_1) and (s_2, t_2) and by creating a path of length $n + 1$ from node t_1 to node s_2 with n new nodes and $n + 1$ new arcs.

Proof. The proof is based on a reduction from the 2-disjoint-paths problem. Consider an instance of the 2-disjoint-paths problem given by a directed graph $G = (N, A)$ and distinct nodes s_1, s_2, t_1, t_2. We construct a graph $G' = (N', A')$ as follows. For $i = 1, 2$, introduce an arc (s_i, t_i). Furthermore, we link node t_1 to node s_2 with $n + 1$ series arcs $(t_1, i_1), (i_1, i_2), \ldots, (i_{n-1}, i_n), (i_n, s_2)$. We associate a cost of zero to all original arcs in A and a cost of one to the new arcs. The construction of graph G' from graph G is shown in Figure 3.

We now suppose that the path P^0 is given as

$$P^0 = (s_1, t_1), (t_1, i_1), (i_1, i_2), \ldots, (i_{n-1}, i_n), (i_n, s_1), (s_2, t_2),$$

and let $K = n + 1$. It is easy to see that the 2-disjoint-paths problem is a "yes" instance if and only if the optimal value of the incremental shortest path problem is $n + 1$ because the incremental shortest path must include the arcs $(t_1, i_1), (i_1, i_2), \ldots, (i_n, s_2)$.

$\square$

4. Adversarial Minimum Spanning Tree Problem

In this section, we consider the *minimum spanning tree* problem. In this problem, we are given an undirected graph $G = (N, A)$ with n nodes and m arcs. Each arc $(i, j) \in A$ has an associated *cost* c_{ij}. The problem is to determine a spanning tree of minimum cost. This problem can be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{(i,j) \in A} x_{ij} \geq n - 1, \\
& \sum_{(i,j) \in A : i, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq N, \\
& x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A.
\end{aligned}
\tag{21}
$$

The binary variable x_{ij} indicates arc (i, j) is in the tree or not, depending on whether $x_{ij} = 0$ or 1, respectively. The first constraint guarantees the existences of at least $n - 1$ arcs. The second set of constraints, referred to it as *subtour-elimination constraints*, ensures that the set of selected arcs does not include any cycle. There is a one-to-one correspondence between feasible solutions of Problem (21) and spanning trees in graph G.

The linear programming relaxation of Problem (21) is known as the *subtour LP*. Edmonds [11] shows that every extreme point of the feasible region of the subtour LP is binary and corresponds to the incidence vector of a spanning tree. In addition, there is a polynomial time separation oracle for the constraints in the subtour LP (see, e.g., [23]). We will rely on this separation algorithm in order to efficiently solve the adversarial problem.

We suppose that each arc $(i, j) \in A$ has an associated interval $[\bar{c}_{ij}, \bar{c}_{ij} + \hat{c}_{ij}]$. Any possible realization of the arc cost c_{ij} lies in this interval. Given an initial spanning tree T^0, one can modify the tree after observing the arc costs. We measure the distance between two spanning trees by the arc inclusion operator; that is, one can build a new spanning tree by adding at most new K arcs. Since each spanning tree has $n - 1$ arcs, this case is equivalent to excluding at most K arcs. It is also equivalent to finding a symmetric difference with at most $2K$ arcs. From the viewpoint of complexity analysis, these three problem variants are equivalent. Accordingly, we just consider the case where a new spanning tree can be built by adding K new arcs.

We will examine the adversarial minimum spanning tree problem under the uncertainty sets $\mathcal{U}_1$ and $\mathcal{U}_2$. We first consider the uncertainty set $\mathcal{U}_2$. The adversarial problem is the problem of finding the Γ most vital arcs, i.e., the Γ arcs whose removal from the graph will lead in the greatest increase in the cost of the minimum spanning tree in the remaining graph. Lin *et al.* [19] show that this problem is NP-hard for arbitrary Γ. Their result immediately implies the following.

Theorem 7. *Under the uncertainty set $\mathcal{U}_2$, the decision variant of the adversarial minimum spanning tree problem is NP-complete.*

We next show that the adversarial minimum spanning tree problem is solvable in polynomial time if the possible range of the cost vector $\mathbf{c}$ is given by $\mathcal{U}_1$. Given a spanning tree T^0 and a vector $\mathbf{c}$ of costs, we define $Z_{\text{Inc}}(T^0, \mathbf{c})$ to be the optimal value of the incremental optimization problem, that is,

$$
\begin{aligned}
Z_{\text{Inc}}(T^0, \mathbf{c}) := \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{(i,j) \notin T^0} x_{ij} \leq K, \\
& \mathbf{x} \in \mathcal{S}.
\end{aligned}
\tag{22}
$$

where $\mathcal{S}$ is the feasible region of Problem (21).

Problem (22) has the integrality property because it is represented as the intersection of two matroids: a uniform matroid and the forest matroid [see 12]. We then apply the Lagrangian relaxation method to relax the first constraint. By associating a Lagrangian multiplier with respect to the first constraint of Problem (22), we obtain the following relaxation problem:

$$
L(\lambda) := \min_{\mathbf{x} \in \mathcal{S}} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} + \lambda \sum_{(i,j) \notin T^0} x_{ij} - \lambda K.
\tag{23}
$$

The binary variables $x_{ij} \in \{0, 1\}$ can be replaced by $x_{ij} \geq 0$ in the Lagrangian relaxation $L(\lambda)$ (note that the constraints $x_{ij} \leq 1, (i, j) \in A$ become redundant due to the subtour-elimination constraints). By strong duality for linear programming, we can write

$$
\begin{aligned}
L(\lambda) = \max \quad & (n-1)w - \sum_{S \subseteq N} (|S| - 1) y_S - \lambda K \\
\text{s.t.} \quad & w - \sum_{S \subseteq N: i, j \in S} y_S \leq c_{ij} && (i, j) \in T^0, \\
& w - \sum_{S \subseteq N: i, j \in S} y_S \leq c_{ij} + \lambda && (i, j) \notin T^0, \\
& y_S \geq 0 && \forall S \subseteq N, \\
& w \geq 0.
\end{aligned}
\tag{24}
$$

Because Problem (22) has the integrality property, $Z_{\text{Inc}}(T^0, \mathbf{c}) = \max_{\lambda \geq 0} L(\lambda)$. The adversarial problem seeks a cost vector $\mathbf{c} \in \mathcal{U}_1$ that maximizes $Z_{\text{Inc}}(T^0, \mathbf{c})$. Let $Z_{\text{Adv}}(T^0)$ denote the optimal value of the adversarial problem, starting

from the initial tree T^0. Following the above discussion, we have

$$Z_{\text{Adv}}(T^0) = \max \quad (n-1)w - \sum_{S \subseteq N}(|S|-1)y_S - \lambda K$$

$$\text{s.t.} \quad w - \sum_{S \subseteq N: i, j \in S} y_S - \delta_{ij} \leq \bar{c}_{ij} \quad (i,j) \in T^0,$$

$$w - \sum_{S \subseteq N: i, j \in S} y_S - \delta_{ij} - \lambda \leq \bar{c}_{ij} \quad (i,j) \notin T^0,$$

$$\sum_{(i,j) \in A} \delta_{ij} \leq \Gamma, \tag{25}$$

$$0 \leq \delta_{ij} \leq \hat{c}_{ij} \quad \forall (i,j) \in A,$$

$$y_S \geq 0 \quad \forall S \subseteq N,$$

$$\lambda, w \geq 0.$$

The dual problem is:

$$Z_{\text{Adv}}(T^0) = \min \quad \sum_{(i,j) \in A} \bar{c}_{ij} x_{ij} + \sum_{(i,j) \in A} \beta_{ij} \hat{c}_{ij} + \theta\Gamma$$

$$\text{s.t.} \quad -x_{ij} + \beta_{ij} + \theta \geq 0 \quad \forall (i,j) \in A,$$

$$\sum_{(i,j) \notin T^0} x_{ij} \leq K, \tag{26}$$

$$\beta_{ij} \geq 0 \quad \forall (i,j) \in A,$$

$$\theta \geq 0,$$

$$\mathbf{x} \in \mathcal{S}_{\text{R}},$$

where $\mathcal{S}_{\text{R}}$ is the feasible region of the subtour LP.

We notice that the set $\mathcal{S}_{\text{R}}$ contains exponentially many constraints. However, as mentioned before, there exists a polynomial time separation oracle for the constraints in $\mathcal{S}_{\text{R}}$. This implies that Problem (26) can solved in polynomial time by the equivalence of optimization and separation [15].

Theorem 8. *Under the uncertainty set $\mathcal{U}_1$, the adversarial minimum spanning tree problem can be solved in polynomial time.*

5. Conclusions

We have presented a robust incremental approach to address uncertainty in optimization problems. We model the case in which the decision maker is allowed to make an incremental change after observing the realization of the uncertain parameters. We addressed the complexity of several optimization problems within this framework. We showed that the robust incremental counterpart of a linear programming problem is solvable in polynomial time. We also established NP-hardness of several robust incremental problems by showing that the adversarial problem or the incremental problem is NP-hard. This puts a lower bound on the complexity of these problems. It is open as to whether the decision versions of the robust incremental problems are Σ_p^2-complete. We remark that if the incremental problem as well as the adversarial problem are in the class NP, then the robust incremental problem is in the class Σ_p^2. Another open issue is the complexity of the robust incremental minimum spanning tree problem.

References

[1] Bar-Noy, A, S. Khuller, B Schieber. 1995. The complexity of finding most vital arcs and nodes. Technical Report CS-TR-3539, Institute for Advanced Studies, University of Maryland, College Park, MD,.

[2] Ben-Tal, A., L. El Ghaoui, A.S. Nemirovski. 2009. *Robust Optimization*. Princeton Series in Applied Mathematics, Princeton University Press.

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

[4] Ben-Tal, A., A. Nemirovski. 1999. Robust solutions of uncertain linear programs. *Operations Research Letters* **25** 1–13.

[5] Bertsimas, D., D. B. Brown, C. Caramanis. 2011. Theory and applications of robust optimization. *SIAM Review* **53**(3) 464–501.

[6] Bertsimas, D., M. Sim. 2003. Robust discrete optimization and network flows. *Mathematical Programming* **98** 49–71.

[7] Bertsimas, D., M. Sim. 2004. The price of robustness. *Operations Research* **54** 35–53.

[8] Birge, J. R., F. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering, Springer.

[9] Büsing, C. 2012. Recoverable robust shortest path problems. *Networks* **59**(1) 181–189.

[10] Şeref, O., R. K. Ahuja, J. B. Orlin. 2009. Incremental network optimization: Theory and algorithms. *Operations Research* **57**(3) 586–594.

[11] Edmonds, J. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards* **71B** 233–240.

[12] Edmonds, J. 1970. Submodular functions, matroids, and certain polyhedra. *Combinatorial Structures and Their Applications* 69–87.

[13] Fortune, S., J. E. Hopcroft, J. Wyllie. 1980. The directed subgraph homeomorphism problem. *Theoretical Computer Science* **10** 111–121.

[14] Goetzmann, K.-S., S. Stiller, C. Telha. 2011. Optimization over integers with robustness in cost and few constraints. Roberto Solis-Oba, Giuseppe Persiano, eds., *WAOA*, *Lecture Notes in Computer Science*, vol. 7164. Springer.

[15] Grötschel, M., L. Lovász, A. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization*, *Algorithms and Combinatorics*, vol. 2. Springer, Berlin.

[16] Israeli, E., R. K. Wood. 2002. Shortest-Path Network Interdiction. *Networks* **40**(2) 97–111.

[17] Kall, P., S. Wallace. 1994. *Stochastic Programming*. John Wiley & Sons.

[18] Liebchen, C., M. E. Lübbecke, R. H. Möhring, S. Stiller. 2009. The concept of recoverable robustness, linear programming recovery, and railway applications. R.K. Ahuja, R.H. Möhring, Chr. Zaroliagis, eds., *Robust and Online Large-Scale Optimization*, *LNCS*, vol. 5868. Springer-Verlag, Berlin, 1–27.

[19] Lin, K. C., M. S. Chern. 1993. The most vital edges in the minimum spanning tree problem. *Information Processing Letters* **45**(1) 25–31.

[20] Meyer, A.R., L.J. Stockmeyer. 1972. The equivalence problem for regular expressions with squaring requires exponential space. *13th Annual Symposium on Switching and Automata Theory*. IEEE, 125–129.

[21] Rocco Sanseverino, C. M., J. E. Ramirez-Marquez. 2010. A bi-objective approach for shortest-path network interdiction. *Computers and Industrial Engineering,* **59**(2) 232–240.

[22] Royset, J. O., R. K. Wood. 2007. Solving the bi-objective maximum-flow network-interdiction problem. *INFORMS Journal on Computing* **19** 175–184.

[23] Schrijver, A. 2003. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

[24] Shapiro, A., D. Dentcheva, A. Ruszczynski. 2009. *Lectures on Stochastic Programming: Modeling and Theory*. No. 9 in MPS-SIAM series on optimization, SIAM, Philadelphia.

[25] Shen, H. 1995. Finding the *k* most vital edges with respect to minimum spanning tree. *Acta Informatica* **36** 405–424.

[26] Stockmeyer, L.J. 1976. The polynomial-time hierarchy. *Theoretical Computer Science* **3**(1) 1–22.

[27] van Hoesel, C. P. M., A. W. J. Kolen, Rinnooy A. H. G. Kan, A. P. M. Wagelmans. 1989. Sensitivity analysis in combinatorial optimization: A bibliography. Technical report 8944/A, Econometric Institute, Erasmus University, Rotterdam, The Netherlands.

[28] Wollmer, R. 1964. Removing arcs from a network. *Transportation Research B* **12** 934–940.

[29] Wood, R. K. 1993. Deterministic network interdiction. *Mathematical and Computer Modelling* **17** 1–18.