Robust Discrete Optimization and Network Flows

Dimitris Bertsimas *　　　Melvyn Sim †

January 2002; revised September 2002

Abstract

We propose an approach to address data uncertainty for discrete optimization and network flow problems that allows controlling the degree of conservatism of the solution, and is computationally tractable both practically and theoretically. In particular, when both the cost coefficients and the data in the constraints of an integer programming problem are subject to uncertainty, we propose a robust integer programming problem of moderately larger size that allows controlling the degree of conservatism of the solution in terms of probabilistic bounds on constraint violation. When only the cost coefficients are subject to uncertainty and the problem is a $0-1$ discrete optimization problem on n variables, then we solve the robust counterpart by solving at most $n+1$ instances of the original problem. Thus, the robust counterpart of a polynomially solvable $0-1$ discrete optimization problem remains polynomially solvable. In particular, robust matching, spanning tree, shortest path, matroid intersection, etc. are polynomially solvable. We also show that the robust counterpart of an NP-hard α-approximable $0-1$ discrete optimization problem, remains α-approximable. Finally, we propose an algorithm for robust network flows that solves the robust counterpart by solving a polynomial number of nominal minimum cost flow problems in a modified network.

*Boeing Professor of Operations Research, Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, E53-363, Cambridge, MA 02139, dbertsim@mit.edu. The research of the author was partially supported by the Singapore-MIT alliance.

†Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, melvyn@mit.edu. The research of the author is supported by a graduate scholarship from the National University of Singapore

1 Introduction

Addressing data uncertainty in mathematical programming models has long been recognized as a central problem in optimization. There are two principal methods that have been proposed to address data uncertainty over the years: (a) stochastic programming, and (b) robust optimization.

As early as the mid 1950s, Dantzig [9] introduced stochastic programming as an approach to model data uncertainty by assuming scenarios for the data occurring with different probabilities. The two main difficulties with such an approach are: (a) Knowing the exact distribution for the data, and thus enumerating scenarios that capture this distribution is rarely satisfied in practice, and (b) the size of the resulting optimization model increases drastically as a function of the number of scenarios, which poses substantial computational challenges.

In recent years a body of literature is developing under the name of robust optimization, in which we optimize against the worst instances that might arise by using a min-max objective. Mulvey et al. [14] present an approach that integrates goal programming formulations with scenario-based description of the problem data. Soyster, in the early 1970s, [17] proposes a linear optimization model to construct a solution that is feasible for all input data such that each uncertain input data can take any value from an interval. This approach, however, tends to find solutions that are over-conservative. Ben-Tal and Nemirovski [3, 4, 5] and El-Ghaoui et al. [11, 12] address the over-conservatism of robust solutions by allowing the uncertainty sets for the data to be ellipsoids, and propose efficient algorithms to solve convex optimization problems under data uncertainty. However, as the resulting robust formulations involve conic quadratic problems (see Ben-Tal and Nemirovski [4]), such methods cannot be directly applied to discrete optimization. Bertsimas and Sim [7] propose a different approach to control the level of conservatism in the solution that has the advantage that leads to a linear optimization model and thus, as we examine in more detail in this paper, can be directly applied to discrete optimization models. We review this work in Section 2.

Specifically for discrete optimization problems, Kouvelis and Yu [13] propose a framework for robust discrete optimization, which seeks to find a solution that minimizes the worst case performance under a set of scenarios for the data. Unfortunately, under their approach, the robust counterpart of many polynomially solvable discrete optimization problems becomes *NP-hard*. A related objective is the minimax-regret approach, which seeks to minimize the worst case loss in objective value that may occur. Again, under the minimax-regret notion of robustness, many of the polynomially solvable discrete optimization problems become NP-hard. Under the minimax-regret robustness approach, Averbakh [2] showed that polynomial solvability is preserved for a specific discrete optimization problem (optimization over a uniform matroid) when each cost coefficient can vary within an interval (interval representation of uncertainty); however, the approach does not seem to generalize to other discrete optimization problems. There have also been research efforts to apply stochastic programming methods to discrete optimization (see for example Schultz et al. [16]), but the computational requirements are even more severe in this case.

Our goal in this paper is to propose an approach to address data uncertainty for discrete optimization and network flow problems that has the following features:

(a) It allows to control the degree of conservatism of the solution;

(b) It is computationally tractable both practically and theoretically.

Specifically, our contributions include:

(a) When both the cost coefficients and the data in the constraints of an integer programming problem are subject to uncertainty, we propose, following the approach in Bertsimas and Sim [7], a robust integer programming problem of moderately larger size that allows to control the degree of conservatism of the solution in terms of probabilistic bounds on constraint violation.

(b) When only the cost coefficients are subject to uncertainty and the problem is a $0-1$ discrete optimization problem on n variables, then we solve the robust counterpart by solving $n+1$ nominal problems. Thus, we show that the robust counterpart of a polynomially solvable $0-1$ discrete optimization problem remains polynomially solvable. In particular, robust matching, spanning tree, shortest path, matroid intersection, etc. are polynomially solvable. Moreover, we show that the robust counterpart of an NP-hard α-approximable $0-1$ discrete optimization problem, remains α-approximable.

(c) When only the cost coefficients are subject to uncertainty and the problem is a minimum cost flow problem, then we propose a polynomial time algorithm for the robust counterpart by solving a collection of minimum cost flow problems in a modified network.

Structure of the paper. In Section 2, we present the general framework and formulation of robust discrete optimization problems. In Section 3, we propose an efficient algorithm for solving robust combinatorial optimization problems. In Section 4, we show that the robust counterpart of an NP-hard $0-1$ α-approximable discrete optimization problem remains α-approximable. In Section 5, we propose an efficient algorithm for robust network flows. In Section 6, we present some experimental findings relating to the computation speed and the quality of robust solutions. Finally, Section 7 contains some remarks with respect to the practical applicability of the proposed methods.

2 Robust Formulation of Discrete Optimization Problems

Let $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{u}$ be n-vectors, let $\boldsymbol{A}$ be an $m \times n$ matrix, and $\boldsymbol{b}$ be an m-vector. We consider the following nominal mixed integer programming (MIP) on a set of n variables, the first k of which are integral:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}'\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u} \\ & x_i \in \mathcal{Z}, \qquad i = 1, \ldots, k, \end{aligned} \tag{1}$$

We assume without loss of generality that data uncertainty affects only the elements of the matrix $\boldsymbol{A}$ and $\boldsymbol{c}$, but not the vector $\boldsymbol{b}$, since in this case we can introduce a new variable x_{n+1}, and write $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}x_{n+1} \le \boldsymbol{0}$, $\boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}, 1 \le x_{n+1} \le 1$, thus augmenting $\boldsymbol{A}$ to include $\boldsymbol{b}$.

In typical applications, we have reasonable estimates for the mean value of the coefficients a_{ij} and its range $\hat{a}_{ij}$. We feel that it is unlikely that we know the exact distribution of these coefficients. Similarly, we have estimates for the cost coefficients c_j and an estimate of its range d_j. Specifically, the model of data uncertainty we consider is as follows:

Model of Data Uncertainty U:

(a) **(Uncertainty for matrix $\boldsymbol{A}$)**: Let $N = \{1, 2, \ldots, n\}$. Each entry a_{ij}, $j \in N$ is modeled as independent, symmetric and bounded random variable (but with unknown distribution) $\tilde{a}_{ij}, j \in N$ that takes values in $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$.

(b) **(Uncertainty for cost vector $\boldsymbol{c}$)**: Each entry c_j, $j \in N$ takes values in $[c_j, c_j + d_j]$, where d_j represents the deviation from the nominal cost coefficient, c_j.

Note that we allow the possibility that $\hat{a}_{ij} = 0$ or $d_j = 0$. Note also that the only assumption that we place on the distribution of the coefficients a_{ij} is that it is symmetric.

2.1 Robust MIP Formulation

For robustness purposes, for every i, we introduce a number Γ_i, $i = 0, 1, \ldots, m$ that takes values in the interval $[0, |J_i|]$, where $J_i = \{j | \ \hat{a}_{ij} > 0\}$. Γ_0 is assumed to be integer, while Γ_i, $i = 1, \ldots, m$ are not necessarily integers.

The role of the parameter Γ_i in the constraints is to adjust the robustness of the proposed method against the level of conservatism of the solution. Consider the ith constraint of the nominal problem $\boldsymbol{a_i'x} \leq b_i$. Let J_i be the set of coefficients a_{ij}, $j \in J_i$ that are subject to parameter uncertainty, i.e., $\tilde{a}_{ij}$, $j \in J_i$ independently takes values according to a symmetric distribution with mean equal to the nominal value a_{ij} in the interval $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. Speaking intuitively, it is unlikely that all of the a_{ij}, $j \in J_i$ will change. Our goal is to be protected against all cases in which up to $\lfloor \Gamma_i \rfloor$ of these coefficients are allowed to change, and one coefficient a_{it} changes by at most $(\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it}$. In other words, we stipulate that nature will be restricted in its behavior, in that only a subset of the coefficients will change in order to adversely affect the solution. We will then guarantee that if nature behaves like this then the robust solution will be feasible deterministically. We will also show that, essentially because the distributions we allow are symmetric, even if more than $\lfloor \Gamma_i \rfloor$ change, then the robust solution will be feasible with very high probability. Hence, we call Γ_i the protection level for the ith constraint.

The parameter Γ_0 controls the level of robustness in the objective. We are interested in finding an optimal solution that optimizes against all scenarios under which a number Γ_0 of the cost coefficients can vary in such a way as to maximally influence the objective. Let $J_0 = \{j | \ d_j > 0\}$. If $\Gamma_0 = 0$, we completely ignore the influence of the cost deviations, while if $\Gamma_0 = |J_0|$, we are considering all possible cost deviations, which is indeed most conservative. In general a higher value of Γ_0 increases the level of robustness at the expense of higher nominal cost.

Specifically, the proposed robust counterpart of Problem (1) is as follows:

$$
\begin{aligned}
\text{minimize} \quad & \boldsymbol{c'x} + \max_{\{S_0 | \ S_0 \subseteq J_0, |S_0| \leq \Gamma_0\}} \left\{ \sum_{j \in S_0} d_j |x_j| \right\} \\
\text{subject to} \quad & \sum_j a_{ij} x_j + \max_{\{S_i \cup \{t_i\} | \ S_i \subseteq J_i, |S_i| \leq \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} |x_j| + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} |x_{t_i}| \right\} \leq b_i, \quad \forall i \\
& \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \\
& x_i \in \mathcal{Z}, \qquad \forall i = 1, \ldots, k.
\end{aligned}
\tag{2}
$$

We next show that the approach in Bertsimas and Sim [7] for linear optimization extends to discrete optimization.

Theorem 1 *Problem (2) has an equivalent MIP formulation as follows:*

$$
\begin{array}{lll}
\text{minimize} & \boldsymbol{c'x} + z_0\Gamma_0 + \sum_{j\in J_0} p_{0j} & \\
\text{subject to} & \displaystyle\sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j\in J_i} p_{ij} \le b_i & \forall i \\
& z_0 + p_{0j} \ge d_j y_j & \forall j \in J_0 \\
& z_i + p_{ij} \ge \hat{a}_{ij} y_j & \forall i \ne 0, j \in J_i \\
& p_{ij} \ge 0 & \forall i, j \in J_i \\
& y_j \ge 0 & \forall j \\
& z_i \ge 0 & \forall i \\
& -y_j \le x_j \le y_j & \forall j \\
& l_j \le x_j \le u_j & \forall j \\
& x_i \in \mathcal{Z} & i = 1, \dots, k.
\end{array}
\tag{3}
$$

Proof : We first show how to model the constraints in (2) as linear constraints. Given a vector $\boldsymbol{x}^*$, we define:

$$
\beta_i(\boldsymbol{x}^*) = \max_{\{S_i \cup \{t_i\}|\ S_i \subseteq J_i, |S_i| \le \lfloor \Gamma_i \rfloor, t_i \in J_i \backslash S_i\}} \left\{ \sum_{j\in S_i} \hat{a}_{ij}|x_j^*| + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it_i}|x_{t_i}^*| \right\}. \tag{4}
$$

This equals to:

$$
\begin{array}{rll}
\beta_i(\boldsymbol{x}^*) = \text{maximize} & \displaystyle\sum_{j\in J_i} \hat{a}_{ij}|x_j^*|z_{ij} & \\
\text{subject to} & \displaystyle\sum_{j\in J_i} z_{ij} \le \Gamma_i & \\
& 0 \le z_{ij} \le 1 & \forall i, j \in J_i.
\end{array}
\tag{5}
$$

Clearly the optimal solution value of Problem (5) consists of $\lfloor \Gamma_i \rfloor$ variables at 1 and one variable at $\Gamma_i - \lfloor \Gamma_i \rfloor$. This is equivalent to the selection of subset $\{S_i \cup \{t_i\}|\ S_i \subseteq J_i, |S_i| \le \lfloor \Gamma_i \rfloor, t_i \in J_i \backslash S_i\}$ with corresponding cost function $\sum_{j\in S_i} \hat{a}_{ij}|x_j^*| + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it_i}|x_{t_i}^*|$. We next consider the dual of Problem (5):

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j\in J_i} p_{ij} + \Gamma_i z_i & \\
\text{subject to} & z_i + p_{ij} \ge \hat{a}_{ij}|x_j^*| & \forall j \in J_i \\
& p_{ij} \ge 0 & \forall j \in J_i \\
& z_i \ge 0 & \forall i.
\end{array}
\tag{6}
$$

By strong duality, since Problem (5) is feasible and bounded for all $\Gamma_i \in [0, |J_i|]$, then the dual problem (6) is also feasible and bounded and their objective values coincide. We have that $\beta_i(\boldsymbol{x}^*)$ is equal to the objective function value of Problem (6).

Similarly we can covert the objective function of Problem (2) to a linear one as follows:

$$\begin{aligned}\beta_0(\boldsymbol{x}^*) &= \max\left\{\textstyle\sum_{j\in S_0} d_j|x_j^*| \ : |S_0| \le \Gamma_0,\ S_0 \subseteq J_0\right\} \\ &= \max\left\{\textstyle\sum_{j\in J_0} d_j|x_j^*|z_{0j} \ : \sum_{j\in J_0} z_{0j} \le \Gamma_0,\ 0 \le z_{0j} \le 1,\ \forall j \in J_0\right\} \\ &= \min\left\{\textstyle\sum_{j\in J_0} p_{0j} + \Gamma_0 z_0 \ : z_0 + p_{0j} \ge d_j|x_j^*|, z_0 \ge 0, p_{0j} \ge 0,\ \forall j \in J_0\right\}\end{aligned} \tag{7}$$

Substituting to Problem (2), we obtain that Problem (2) is equivalent to Problem (3). ■

While the original Problem (1) involves n variables and m constraints, its robust counterpart Problem (3) has $2n+m+l$ variables, where $l = \sum_{i=0}^m |J_i|$ is the number of uncertain coefficients, and $2n+m+l$ constraints.

As we discussed, if less than $\lfloor \Gamma_i \rfloor$ coefficients a_{ij}, $j \in J_i$ participating in the ith constraint vary, then the robust solution will be feasible deterministically. We next show that even if more than $\lfloor \Gamma_i \rfloor$ change, then the robust solution will be feasible with very high probability.

Theorem 2 *(Bertsimas and Sim [7]) Let $\boldsymbol{x}^*$ be an optimal solution of Problem (3).*

(a) *Suppose that the data in matrix $\boldsymbol{A}$ are subject to the model of data uncertainty U, the probability that the ith constraint is violated satisfies:*

$$\Pr\left(\sum_j \tilde{a}_{ij}x_j^* > b_i\right) \le B(n,\Gamma_i) = \frac{1}{2^n}\left\{(1-\mu)\sum_{l=\lfloor\nu\rfloor}^{n}\binom{n}{l} + \mu\sum_{l=\lfloor\nu\rfloor+1}^{n}\binom{n}{l}\right\}, \tag{8}$$

where $n = |J_i|$, $\nu = \frac{\Gamma_i+n}{2}$ and $\mu = \nu - \lfloor\nu\rfloor$. Moreover, the bound is tight.

(b) *The bound (8) satisfies*

$$B(n,\Gamma_i) \le (1-\mu)C(n,\lfloor\nu\rfloor) + \sum_{l=\lfloor\nu\rfloor+1}^{n} C(n,l), \tag{9}$$

where

$$C(n,l) = \begin{cases} \dfrac{1}{2^n}, & \text{if } l\text{=0 or } l\text{=}n, \\ \dfrac{1}{\sqrt{2\pi}}\sqrt{\dfrac{n}{(n-l)l}}\exp\left(n\log\left(\dfrac{n}{2(n-l)}\right) + l\log\left(\dfrac{n-l}{l}\right)\right), & \text{otherwise.}\end{cases} \tag{10}$$

(c) *For $\Gamma_i = \theta\sqrt{n}$,*

$$\lim_{n\to\infty} B(n,\Gamma_i) = 1 - \Phi(\theta), \tag{11}$$

where

$$\Phi(\theta) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\theta}\exp\left(-\frac{y^2}{2}\right)dy$$

is the cumulative distribution function of a standard normal.

Remarks:

(a) The bound (8) is independent of $\boldsymbol{x}^*$.

(b) While Bound (8) is best possible it poses computational difficulties in evaluating the sum of combination functions for large n. For this reason, we have calculated Bound (9), which is simple to compute and, as Bertsimas and Sim [7] show, very tight.

(c) Eq. (11) is a formal asymptotic theorem that applies when $\Gamma_i = \theta\sqrt{n}$. We can use the De Moivre-Laplace approximation of the Binomial distribution to obtain the approximation

$$B(n, \Gamma_i) \approx 1 - \Phi\left(\frac{\Gamma_i - 1}{\sqrt{n}}\right), \tag{12}$$

that applies, even when Γ_i does not scale as $\theta\sqrt{n}$.

(d) We make no theoretical claims regarding suboptimality given that we made no probabilistic assumptions on the cost coefficients. In Section 6.1, we apply these bounds in the context of the zero-one knapsack problem.

3 Robust Combinatorial Optimization

Combinatorial optimization is an important class of discrete optimization whose decision variables are binary, that is $\boldsymbol{x} \in X \subseteq \{0,1\}^n$. In this section, the nominal combinatorial optimization problem we consider is:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}'\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x} \in X. \end{aligned} \tag{13}$$

We are interested in the class of problems where each entry $\tilde{c}_j,\ j \in N = \{1, 2, \ldots, n\}$ takes values in $[c_j, c_j + d_j]$, $d_j \geq 0,\ j \in N$, but the set X is fixed. We would like to find a solution $\boldsymbol{x} \in X$ that minimizes the maximum cost $\boldsymbol{c}'\boldsymbol{x}$ such that at most Γ of the coefficients $\tilde{c}_j$ are allowed to change:

$$\begin{aligned} Z^* = \text{minimize} \quad & \boldsymbol{c}'\boldsymbol{x} + \max_{\{S|\ S \subseteq N, |S| \leq \Gamma\}} \sum_{j \in S} d_j x_j \\ \text{subject to} \quad & \boldsymbol{x} \in X. \end{aligned} \tag{14}$$

Without loss of generality, we assume that the indices are ordered in such that $d_1 \geq d_2 \geq \ldots \geq d_n$. We also define $d_{n+1} = 0$ for notational convenience. Examples of such problems include the shortest path, the minimum spanning tree, the minimum assignment, the traveling salesman, the vehicle routing and matroid intersection problems. Data uncertainty in the context of the vehicle routing problem for example, captures the variability of travel times in some of the links of the network.

In the context of scenario based uncertainty, finding an optimally robust solution involves solving the problem (for the case that only two scenarios for the cost vectors $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ are known):

$$\begin{aligned} \text{minimize} \quad & \max(\boldsymbol{c_1'x}, \boldsymbol{c_2'x}) \\ \text{subject to} \quad & \boldsymbol{x} \in X. \end{aligned}$$

For many classical combinatorial problems (for example the shortest path problem), finding such a robust solution is NP-hard, even if minimizing $\boldsymbol{c_i'x}$ subject to $\boldsymbol{x} \in X$ is polynomially solvable (Kouvelis and Yu [13]).

Clearly the robust counterpart of an NP-hard combinatorial optimization problem is NP-hard. We next show that surprisingly, the robust counterpart of a polynomially solvable combinatorial optimization problem is also polynomially solvable.

3.1 Algorithm for Robust Combinatorial Optimization Problems

In this section, we show that we can solve Problem (14) by solving at most $n+1$ nominal problems $\min\ \boldsymbol{f}_i'\boldsymbol{x}$, subject to $\boldsymbol{x} \in X$, for $i = 1, \ldots, n+1$.

Theorem 3 *Problem (14) can be solved by solving the $n+1$ nominal problems:*

$$Z^* = \min_{l=1,\ldots,n+1} G^l, \tag{15}$$

where for $l = 1, \ldots, n+1$:

$$\begin{aligned} G^l = \Gamma d_l + \min \quad & \left(\boldsymbol{c'x} + \sum_{j=1}^{l} (d_j - d_l)\, x_j \right) \\ \text{subject to} \quad & \boldsymbol{x} \in X. \end{aligned} \tag{16}$$

Proof : Problem (14) can be rewritten as follows:

$$\begin{aligned} Z^* = \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \max \quad \right. & \left. \sum_{j \in N} d_j x_j u_j \right) \\ \text{subject to} \quad & 0 \le u_j \le 1, \qquad j \in N \\ & \sum_{j \in N} u_j \le \Gamma. \end{aligned}$$

Given a fixed $\boldsymbol{x} \in X$, we consider the inner maximization problem and formulate its dual. Applying strong duality to this problem we obtain:

$$\begin{aligned} Z^* = \min_{\mathbf{x} \in X} \boldsymbol{c'x} + \min \quad & \left(\Gamma\theta + \sum_{j \in N} y_j \right) \\ \text{subject to} \quad & y_j + \theta \ge d_j x_j, \qquad j \in N \\ & y_j, \theta \ge 0, \end{aligned}$$

which can be rewritten as:

$$
\begin{aligned}
Z^* = \min \quad & \boldsymbol{c'x} + \Gamma\theta + \sum_{j \in N} y_j \\
\text{subject to} \quad & y_j + \theta \geq d_j x_j, \qquad j \in N \\
& y_j, \theta \geq 0, \\
& \boldsymbol{x} \in X.
\end{aligned}
\tag{17}
$$

Clearly an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \theta^*)$ of Problem (17) satisfies:

$$
y_j^* = \max(d_j x_j^* - \theta^*, 0),
$$

and therefore,

$$
Z^* = \min_{\mathbf{x} \in X, \theta \geq 0} \left(\Gamma\theta + \boldsymbol{c'x} + \sum_{j \in N} \max(d_j x_j - \theta, 0) \right).
$$

Since $X \subset \{0,1\}^n$,

$$
\max(d_j x_j - \theta, 0) = \max(d_j - \theta, 0)\; x_j, \tag{18}
$$

Hence, we obtain

$$
Z^* = \min_{\mathbf{x} \in X, \theta \geq 0} \left(\Gamma\theta + \boldsymbol{c'x} + \sum_{j \in N} \max(d_j - \theta, 0) x_j \right). \tag{19}
$$

In order to find the optimal value for θ we decompose $\Re^+$ into the intervals $[0, d_n]$, $[d_n, d_{n-1}], \ldots, [d_2, d_1]$ and $[d_1, \infty)$. Then, recalling that $d_{n+1} = 0$, we obtain

$$
\sum_{j \in N} \max(d_j - \theta, 0) x_j = \begin{cases} \displaystyle\sum_{j=1}^{l-1} (d_j - \theta) x_j, & \text{if } \theta \in [d_l, d_{l-1}], \quad l = n+1, \ldots, 2, \\ 0, & \text{if } \theta \in [d_1, \infty). \end{cases}
$$

Therefore, $Z^* = \min\limits_{l=1,\ldots,n+1} Z^l$, where for $l = 1, \ldots, n+1$:

$$
Z^l = \min_{\mathbf{x} \in X, \theta \in [d_l, d_{l-1}]} \left(\Gamma\theta + \boldsymbol{c'x} + \sum_{j=1}^{l-1} (d_j - \theta) x_j \right),
$$

where the sum for $l = 1$ is equal to zero. Since we are optimizing a linear function of θ over the interval $[d_l, d_{l-1}]$, the optimal is obtained for $\theta = d_l$ or $\theta = d_{l-1}$, and thus for $l = 1, \ldots, n+1$:

$$
\begin{aligned}
Z^l &= \min \left(\Gamma d_l + \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \sum_{j=1}^{l-1} (d_j - d_l) x_j \right), \Gamma d_{l-1} + \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \sum_{j=1}^{l-1} (d_j - d_{l-1}) x_j \right) \right) \\
&= \min \left(\Gamma d_l + \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \sum_{j=1}^{l} (d_j - d_l) x_j \right), \Gamma d_{l-1} + \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \sum_{j=1}^{l-1} (d_j - d_{l-1}) x_j \right) \right).
\end{aligned}
$$

Thus,

$$Z^* = \min\left(\Gamma d_1 + \min_{\mathbf{x}\in X} \boldsymbol{c'x}, \dots, \Gamma d_l + \min_{\mathbf{x}\in X}\left(\boldsymbol{c'x} + \sum_{j=1}^{l}(d_j - d_l)x_j\right), \dots, \min_{\mathbf{x}\in X}\left(\boldsymbol{c'x} + \sum_{j=1}^{n} d_j x_j\right)\right).$$

■

Remark: Note that we critically used the fact that the nominal problem is a 0-1 discrete optimization problem, i.e., $X \subseteq \{0,1\}^n$, in Eq. (18). For general integer optimization problems Eq. (18) does not apply.

Theorem 3 leads to the following algorithm.

Algorithm A

1. For $l = 1, \dots, n+1$ solve the $n+1$ nominal problems Eqs. (16):

$$G^l = \Gamma d_l + \min_{\mathbf{x}\in X}\left(\boldsymbol{c'x} + \sum_{j=1}^{l}(d_j - d_l)x_j\right),$$

   and let $\boldsymbol{x}^l$ be an optimal solution of the corresponding problem.

2. Let $l^* = \arg\min_{l=1,\dots,n+1} G^l$.

3. $Z^* = G^{l^*}$; $\boldsymbol{x^*} = \boldsymbol{x}^{l^*}$.

Note that Z^l is not in general equal to G^l. If f is the number of distinct values among $d_1, \dots, d_n$, then it is clear that Algorithm A solves $f+1$ nominal problems, since if $d_l = d_{l+1}$, then $G^l = G^{l+1}$. In particular, if all $d_j = d$ for all $j = 1, \dots, n$, then Algorithm A solves only two nominal problems. Thus, if τ is the time to solve one nominal problem, Algorithm A solves the robust counterpart in $(f+1)\tau$ time, thus preserving the polynomial solvability of the nominal problem. In particular, Theorem 3 implies that the robust counterpart of many classical 0-1 combinatorial optimization problems like the minimum spanning tree, the minimum assignment, minimum matching, shortest path and matroid intersection, are polynomially solvable.

4 Robust Approximation Algorithms

In this section, we show that if the nominal combinatorial optimization problem (13) has an α-approximation polynomial time algorithm, then the robust counterpart Problem (14) with optimal solution value Z^* is also α-approximable. Specifically, we assume that there exists a polynomial time

Algorithm H for the nominal problem (13), that returns a solution with an objective Z_H: $Z \leq Z_H \leq \alpha Z$, $\alpha \geq 1$.

The proposed algorithm for the robust Problem (14) is to utilize Algorithm H in Algorithm A, instead of solving the nominal instances exactly. The proposed algorithm is as follows:

Algorithm B

1. For $l = 1, \ldots, n+1$ find an α-approximate solution $\boldsymbol{x}_H^l$ using Algorithm H for the nominal problem:

$$G^l - \Gamma d_l = \min_{\mathbf{x} \in X} \left(\boldsymbol{c'x} + \sum_{j=1}^{l} (d_j - d_l) x_j \right). \tag{20}$$

2. For $l = 1, \ldots, n+1$, let

$$Z_H^l = \boldsymbol{c'x}_H^l + \max_{\{S \mid \, S \subseteq N, |S| \leq \Gamma\}} \sum_{j \in S} d_j (\boldsymbol{x}_H^l)_j.$$

3. Let $l^* = \arg \min_{l=1,\ldots,n+1} Z_H^l$.

4. $Z_B = Z_H^{l^*}$; $\boldsymbol{x}^B = \boldsymbol{x}_H^{l^*}$.

Theorem 4 *Algorithm B yields a solution $\boldsymbol{x}^B$ with an objective value Z_B that satisfies:*

$$Z^* \leq \; Z_B \leq \; \alpha Z^*.$$

Proof : Since Z^* is the optimal objective function value of the robust problem, clearly $Z^* \leq \; Z_B$. Let l the index such that $Z^* = G^l$ in Theorem 3. Let $\boldsymbol{x}_H^l$ be an α-optimal solution for Problem (20). Then, we have

$$\begin{aligned}
Z_B &\leq Z_H^l \\
&= \boldsymbol{c'x}_H^l + \max_{\{S \mid \, S \subseteq N, |S| \leq \Gamma\}} \sum_{j \in S} d_j (\boldsymbol{x}_H^l)_j \\
&= \min_{\theta \geq 0} \left\{ \boldsymbol{c'x}_H^l + \sum_{j \in N} \max(d_j - \theta, 0) (\boldsymbol{x}_H^l)_j + \Gamma \theta \right\} && \text{(from Eq. (19))} \\
&\leq \boldsymbol{c'x}_H^l + \sum_{j=1}^{l} (d_j - d_l)(\boldsymbol{x}_H^l)_j + \Gamma d_l \\
&\leq \alpha (G^l - \Gamma d_l) + \Gamma d_l && \text{(from Eq. (20))} \\
&\leq \alpha G^l && \text{(since } \alpha \geq 1\text{)} \\
&= \alpha Z^*.
\end{aligned}$$

■

Remark : Note that Algorithm A is a special case of Algorithm B for $\alpha = 1$. Note that it is critical to have an α-approximation algorithm for all nominal instances (20). In particular, if the nominal problem is the traveling salesman problem under triangle inequality, which can be approximated within $\alpha = 3/2$, Algorithm B is not an α-approximation algorithm for the robust counterpart, as the instances (20) may not satisfy the triangle inequality.

5 Robust Network Flows

In this section, we apply the methods of Section 3 to show that robust minimum cost flows can also be solved by solving a collection of modified nominal minimum cost flows. Given a directed graph $G = (\mathcal{N}, \mathcal{A})$, the minimum cost flow is defined as follows:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} & \\ \text{subject to} & \displaystyle\sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = b_i & \forall i \in \mathcal{N} \\ & 0 \le x_{ij} \le u_{ij} & \forall (i,j) \in \mathcal{A}. \end{array} \tag{21}$$

Let X be the set of feasible solutions of Problem (21).

We are interested in the class of problems in which each entry $\tilde{c}_{ij}$, $(i,j) \in \mathcal{A}$ takes values in $[c_{ij}, c_{ij} + d_{ij}]$, $d_{ij}, c_{ij} \ge 0, (i,j) \in \mathcal{A}$. From Eq. (14) the robust minimum cost flow problem is:

$$\begin{array}{ll} Z^* = \min & \boldsymbol{c'x} + \displaystyle\max_{\{S|\ S\subseteq\mathcal{A},|S|\le\Gamma\}} \sum_{(i,j)\in S} d_{ij}x_{ij} \\ \text{subject to} & \boldsymbol{x} \in X. \end{array} \tag{22}$$

From Eq. (17) we obtain that Problem (22) is equivalent to solving the following problem:

$$Z^* = \min_{\theta\ge 0} Z(\theta), \tag{23}$$

where

$$\begin{array}{lll} Z(\theta) = \Gamma\theta + \min & \boldsymbol{c'x} + \displaystyle\sum_{(i,j)\in\mathcal{A}} p_{ij} & \\ \text{subject to} & p_{ij} \ge d_{ij}x_{ij} - \theta & \forall (i,j) \in \mathcal{A} \\ & p_{ij} \ge 0 & \forall (i,j) \in \mathcal{A} \\ & \boldsymbol{x} \in X. & \end{array} \tag{24}$$

We next show that for a fixed $\theta \ge 0$, we can solve Problem (24) as a network flow problem.

Theorem 5 *For a fixed $\theta \ge 0$, Problem (24) can be solved as a network flow problem.*

Proof :

We eliminate the variables p_{ij} from Formulation (24) and obtain:

$$\begin{aligned} Z(\theta) = \Gamma\theta + \min \quad & \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} d_{ij} \max\left(x_{ij} - \frac{\theta}{d_{ij}}, 0\right) \\ \text{subject to} \quad & \mathbf{x} \in X. \end{aligned} \tag{25}$$

For every arc $(i,j) \in \mathcal{A}$, we introduce nodes i' and j' and replace the arc (i,j) with arcs (i,i'), (i',j'), (j',j) and (i',j) with the following costs and capacities (see also Figure 1):

$$\begin{aligned} c_{ii'} &= c_{ij} & u_{ii'} &= u_{ij} \\ c_{j'j} &= 0 & u_{j'j} &= \infty \\ c_{i'j} &= 0 & u_{i'j} &= \frac{\theta}{d_{ij}} \\ c_{i'j'} &= d_{ij} & u_{i'j'} &= \infty. \end{aligned}$$

Let $G' = (\mathcal{N}', \mathcal{A}')$ be the new direceted graph. We show that solving a linear minimum cost flow

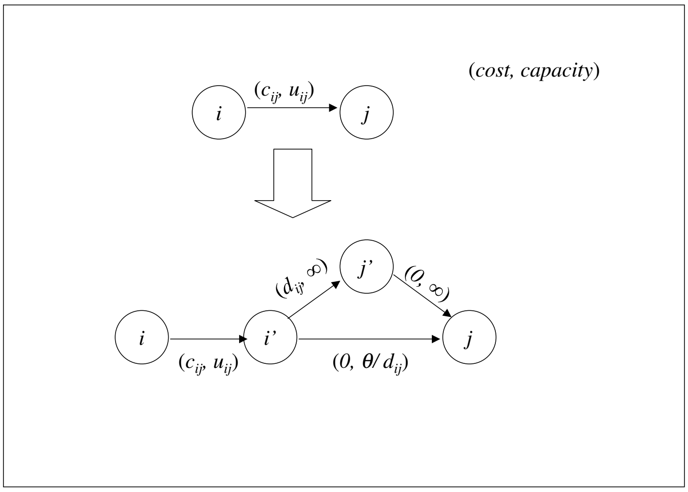

Figure 1: Conversion of arcs with cost uncertainties.

problem with data as above, leads to the solution of Problem (25). Consider an optimal solution of Problem (25). If $x_{ij} \leq \theta/d_{ij}$ for a given arc $(i,j) \in \mathcal{A}$, then the flow x_{ij} will be routed along the arcs (i,i') and (i',j) an the total contribution to cost is

$$c_{ii'}x_{ij} + c_{i'j}x_{ij} = c_{ij}x_{ij}.$$

If, however, $x_{ij} \geq \theta/d_{ij}$, then the flow x_{ij} will be routed along the arcs (i, i'), then θ/d_{ij} will be routed along arc (i', j), and the excess $x_{ij} - (\theta/d_{ij})$ is routed through the arcs (i', j') and (j', j). The total contribution to cost is

$$c_{ii'}x_{ij} + c_{i'j}\frac{\theta}{d_{ij}} + c_{i'j'}\left(x_{ij} - \frac{\theta}{d_{ij}}\right) + c_{j'j}\left(x_{ij} - \frac{\theta}{d_{ij}}\right) =$$

$$c_{ij}x_{ij} + d_{ij}\left(x_{ij} - \frac{\theta}{d_{ij}}\right).$$

In both cases the contribution to cost matches the objective function value in Eq. (25). ■

Without loss of generality, we can assume that all the capacities $u_{ij},\ (i,j) \in \mathcal{A}$ are finitely bounded. Then, clearly $\theta \leq \overline{\theta} = \max\{u_{ij}d_{ij}\ : (i,j) \in \mathcal{A}\}$. Theorem 5 shows that the robust counterpart of the minimum cost flow problem can be converted to a minimum cost flow problem in which capacities on the arcs are linear functions of θ. Srinivasan and Thompsom [18] proposed a simplex based method for solving such parametric network flow problems for all values of the parameter $\theta \in [0, \overline{\theta}]$. Using this method, we can obtain the complete set of robust solutions for $\Gamma \in [0, |\mathcal{A}|]$. However, while the algorithm may be practical, it is not polynomial. We next provide a polynomial time algorithm. We first establish some properties of the function $Z(\theta)$.

Theorem 6 **(a)** *$Z(\theta)$ is a convex function.*
(b) *For all $\theta_1, \theta_2 \geq 0$, we have*

$$|Z(\theta_1) - Z(\theta_2)| \leq |\mathcal{A}||\theta_1 - \theta_2|. \tag{26}$$

Proof :
(a) Let $(\boldsymbol{x_1}, \boldsymbol{p_1})$ and $(\boldsymbol{x_2}, \boldsymbol{p_2})$ be optimal solutions to Problem (24) with $\theta = \theta_1$ and $\theta = \theta_2$ respectively. Clearly, since the feasible region is convex, for all $\lambda \in [0,1]$, $(\lambda\boldsymbol{x_1} + (1-\lambda)\boldsymbol{x_2}, \lambda\boldsymbol{p_1} + (1-\lambda)\boldsymbol{p_2})$ is feasible to the problem with $\theta = \lambda\theta_1 + (1-\lambda)\theta_2$. Therefore,

$$\lambda Z(\theta_1) + (1-\lambda)Z(\theta_2) = \boldsymbol{c}'(\lambda\boldsymbol{x_1} + (1-\lambda)\boldsymbol{x_2}) + \boldsymbol{e}'(\lambda\boldsymbol{p_1} + (1-\lambda)\boldsymbol{p_2}) + \Gamma(\lambda\theta_1 + (1-\lambda)\theta_2) \geq Z(\lambda\theta_1 + (1-\lambda)\theta_2),$$

where $\boldsymbol{e}$ is a vector of ones.
(b) By introducing Lagrange multiplies $\boldsymbol{r}$ to the first set of constraints of Problem (24), we obtain:

$$\begin{aligned}
Z(\theta) &= \max_{\mathbf{r}\geq\mathbf{0}} \min_{\mathbf{x}\in X, \mathbf{p}\geq\mathbf{0}} \left\{ \Gamma\theta + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} p_{ij} + \sum_{(i,j)\in\mathcal{A}} r_{ij}(d_{ij}x_{ij} - p_{ij} - \theta) \right\} \\
&= \max_{\mathbf{r}\geq\mathbf{0}} \min_{\mathbf{x}\in X, \mathbf{p}\geq\mathbf{0}} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})\theta + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} p_{ij}(1 - r_{ij}) + \sum_{(i,j)\in\mathcal{A}} r_{ij}d_{ij}x_{ij} \right\} \\
&= \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})\theta + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij}d_{ij}x_{ij} \right\},
\end{aligned} \tag{27}$$

where Eq. (27) follows from the fact that $\min_{\mathbf{p}\geq\mathbf{0}}\left\{\sum_{(i,j)\in\mathcal{A}} p_{ij}(1-r_{ij})\right\}$ is unbounded if any $r_{ij} > 1$ and equals to zero for $\mathbf{0} \leq \boldsymbol{r} \leq \mathbf{e}$. Without loss of generality, let $\theta_1 > \theta_2 \geq 0$. For $\mathbf{0} \leq \boldsymbol{r} \leq \mathbf{e}$, we have

$$-|\mathcal{A}| \leq \Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij} \leq |\mathcal{A}|.$$

Thus,

$$\begin{aligned}
Z(\theta_1) &= \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})\theta_1 + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij} d_{ij} x_{ij} \right\} \\
&= \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})(\theta_2 + (\theta_1 - \theta_2)) + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij} d_{ij} x_{ij} \right\} \\
&\leq \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})\theta_2 + |\mathcal{A}|(\theta_1 - \theta_2) + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij} d_{ij} x_{ij} \right\} \\
&= Z(\theta_2) + |\mathcal{A}|(\theta_1 - \theta_2).
\end{aligned}$$

Similarly,

$$\begin{aligned}
Z(\theta_1) &= \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})(\theta_2 + (\theta_1 - \theta_2)) + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij} d_{ij} x_{ij} \right\} \\
&\geq \max_{\mathbf{0}\leq\mathbf{r}\leq\mathbf{e}} \min_{\mathbf{x}\in X} \left\{ (\Gamma - \sum_{(i,j)\in\mathcal{A}} r_{ij})\theta_2 - |\mathcal{A}|(\theta_1 - \theta_2) + \boldsymbol{c'x} + \sum_{(i,j)\in\mathcal{A}} r_{ij} d_{ij} x_{ij} \right\} \\
&= Z(\theta_2) - |\mathcal{A}|(\theta_1 - \theta_2).
\end{aligned}$$

■

We next show that the robust minimum cost flow problem (22) can be solved by solving a polynomial number of network flow problems.

Theorem 7 *For any fixed $\Gamma \leq |\mathcal{A}|$ and every $\epsilon > 0$, we can find a solution $\hat{\mathbf{x}} \in X$ with robust objective value*

$$\hat{Z} = \boldsymbol{c}'\hat{\mathbf{x}} + \max_{\{S|\ S\subseteq\mathcal{A},|S|\leq\Gamma\}} \sum_{(i,j)\in S} d_{ij}\hat{x}_{ij}$$

such that

$$Z^* \leq \hat{Z} \leq (1+\epsilon)Z^*$$

by solving $2\lceil\log_2(|\mathcal{A}|\overline{\theta}/\epsilon)\rceil + 3$ network flow problems, where $\overline{\theta} = \max\{u_{ij}d_{ij}\ : (i,j) \in \mathcal{A}\}$.

Proof : Let $\theta^* \geq 0$ be such that $Z^* = Z(\theta^*)$. Since $Z(\theta)$ is a convex function (Theorem 6(a)), we use binary search to find a $\hat{\theta}$ such that

$$|\hat{\theta} - \theta^*| \leq \frac{\overline{\theta}}{2^k},$$

by solving $2k+3$ minimum cost flow problems of the type described in Theorem 5. We first need to evaluate $Z(0), Z(\overline{\theta}/2), Z(\overline{\theta})$, and then we need two extra points $Z(\overline{\theta}/4)$ and $Z(3\overline{\theta}/4)$ in order to decide whether θ^* belongs in the interval $[0, \overline{\theta}/2]$ or $[\overline{\theta}/2, \overline{\theta}]$ or $[\overline{\theta}/4, 3\overline{\theta}/4]$. From then on, we need two extra evaluations in order to halve the interval θ^* can belong to.

From Theorem 6(b)

$$|Z(\hat{\theta}) - Z(\theta^*)| \leq |\mathcal{A}||\hat{\theta} - \theta^*| \leq |\mathcal{A}|\frac{\overline{\theta}}{2^k} \leq \epsilon,$$

for $k = \lceil \log_2(|\mathcal{A}|\overline{\theta}/\epsilon) \rceil$. Note that $\hat{\mathbf{x}}$ is the flow corresponding to the nominal network flow problem for $\theta = \hat{\theta}$. ■

6 Experimental Results

In this section we consider concrete discrete optimization problems and solve the robust counterparts.

6.1 The Robust Knapsack Problem

The zero-one nominal knapsack problem is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N} c_i x_i \\ \text{subject to} \quad & \sum_{i \in N} w_i x_i \leq b \\ & \boldsymbol{x} \in \{0,1\}^n. \end{aligned}$$

We assume that the weights $\tilde{w}_i$ are uncertain, independently distributed and follow symmetric distributions in $[w_i - \delta_i, w_i + \delta_i]$. The objective value vector $\boldsymbol{c}$ is not subject to data uncertainty. An application of this problem is to maximize the total value of goods to be loaded on a cargo that has strict weight restrictions. The weight of the individual item is assumed to be uncertain, independent of other weights and follows a symmetric distribution. In our robust model, we want to maximize the total value of the goods but allowing a maximum of 1% chance of constraint violation.

The robust Problem (2) is as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N} c_i x_i \\ \text{subject to} \quad & \sum_{i \in N} w_i x_i + \max_{\{S \cup \{t\} \mid \; S \subseteq N, |S| = \lfloor \Gamma \rfloor, t \in N \setminus S\}} \left\{ \sum_{j \in S} \delta_j x_j + (\Gamma - \lfloor \Gamma \rfloor)\delta_t x_t \right\} \leq b \\ & \boldsymbol{x} \in \{0,1\}^n. \end{aligned}$$

Γ	Violation Probability	Optimal Value	Reduction
0	1	5592	0%
2.8	4.49×10^{-1}	5585	0.13%
36.8	5.71×10^{-3}	5506	1.54%
82.0	5.04×10^{-9}	5408	3.29%
200	0	5283	5.50%

Table 1: Robust Knapsack Solutions.

For this experiment, we solve Problem (3) using CPLEX 7.0 for a random knapsack problem of size, $|N| = 200$. We set the capacity limit, b to 4000, the nominal weight, w_i being randomly chosen from the set $\{20, 21, \ldots, 29\}$ and the cost c_i randomly chosen from the set $\{16, 17, \ldots, 77\}$. We set the weight uncertainty δ_i to equal 10% of the nominal weight. The time to solve the robust discrete problems to optimality using CPLEX 7.0 on a Pentium II 400 PC ranges from 0.05 to 50 seconds.

Under zero protection level, $\Gamma = 0$, the optimal value is $5,592$. However, with full protection, $\Gamma = 200$, the optimal value is reduced by 5.5% to $5,283$. In Table 1, we present a sample of the objective function value and the probability bound of constraint violation computed from Eq. (8). It is interesting to note that the optimal value is marginally affected when we increase the protection level. For instance, to have a probability guarantee of at most 0.57% chance of constraint violation, we only reduce the objective by 1.54%. It appears that in this example we do not heavily penalize the objective function value in order to protect ourselves against constraint violation.

We repeated the experiment twenty times and in Figure 2 we report the tradeoff between robustness and optimality for all twenty problems. We observe that by allowing a profit reduction of 2%, we can make the probability of constraint violation smaller than 10^{-3}. Moroever, the conclusion did not seem to depend a lot on the specific instance we generated.

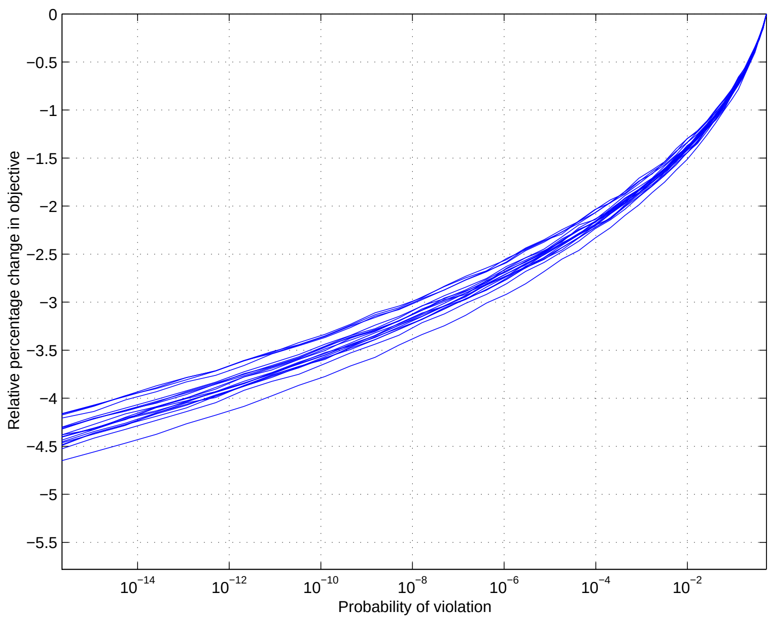

Figure 2: The tradeoff between robustness and optimality in twenty instances of the 0-1 knapsack problem.

6.2 Robust Sorting

We consider the problem of minimizing the total cost of selecting k items out of a set of n items that can be expressed as the following integer programming problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in N} c_i x_i \\ \text{subject to} \quad & \sum_{i\in N} x_i = k \\ & \boldsymbol{x} \in \{0,1\}^n. \end{aligned} \tag{28}$$

In this problem, the cost components are subjected to uncertainty. If the model is deterministic, we can easily solve the problem in $O(n \log n)$ by sorting the costs in ascending order and choosing the first k items. However, under the influence of data uncertainty, we will illustrate empirically that the deterministic model could lead to large deviations when the cost components are subject to uncertainty. Under our proposed Problem (14), we solve the following problem,

$$\begin{aligned} Z^*(\Gamma) = \quad \text{minimize} \quad & \boldsymbol{c'x} + \max_{\{S|\ S\subseteq J, |S|\le\Gamma\}} \sum_{j\in S} d_j x_j \\ \text{subject to} \quad & \sum_{i\in N} x_i = k \\ & \boldsymbol{x} \in \{0,1\}^n. \end{aligned} \tag{29}$$

We experiment with a problem of size $|N| = 200$ and $k = 100$. The cost and deviation components, c_j and d_j are uniformly distributed in $[50, 200]$ and $[20, 200]$ respectively. Since only k items will be selected, the robust solution for $\Gamma > k$ is the same as when $\Gamma = k$. Hence, Γ takes integral values from $[0, k]$. By varying Γ, we will illustrate empirically that we can control the deviation of the objective value under the influence of cost uncertainty.

We solve Problem (29) in two ways. First using Algorithm A, and second solving Problem (3):

$$
\begin{aligned}
\text{minimize} \quad & \boldsymbol{c'x} + z\Gamma + \sum_{j\in N} p_j \\
\text{subject to} \quad & z + p_j \geq d_j x_j \qquad \forall j \in N \\
& \sum_{i\in N} x_i = k \\
& z \geq 0 \\
& p_j \geq 0 \\
& \boldsymbol{x} \in \{0,1\}^n.
\end{aligned} \tag{30}
$$

Algorithm A was able to find the robust solution for all $\Gamma \in \{0, \dots k\}$ in less than a second. The typical running time using CPLEX 7.0 to solve Problem (30) for only one of the Γ ranges from 30 to 80 minutes, which underscores the effectiveness of Algorithm A.

We let $\boldsymbol{x}(\Gamma)$ be an optimal solution to the robust model, with parameter Γ and define $\overline{Z}(\Gamma) = \boldsymbol{c'x}(\Gamma)$ as the nominal cost in the absence of any cost deviations. To analyze the robustness of the solution, we simulate the distribution of the objective by subjecting the cost components to random perturbations. Under the simulation, each cost component independently deviates with probability p from the nominal value c_j to $c_j + d_j$. In Table 2, we report $\overline{Z}(\Gamma)$ and the standard deviation $\sigma(\Gamma)$ found in the simulation for $p = 0.2$ (we generated 20,000 instances to evaluate $\sigma(\Gamma)$).

Table 2 suggests that as we increase Γ, the standard deviation of the objective, $\sigma(\Gamma)$ decreases, implying that the robustness of the solution increases, and $\overline{Z}(\Gamma)$ increases. Varying Γ we can find the tradeoff between the variability of the objective and the increase in nominal cost. Note that the robust formulation does not explicitly consider standard deviation. We chose to represent robustness in the numerical results with standard deviation of the objective, since standard deviation is the standard measure of variability and it has intuitive appeal.

In Figure 3 we report the cumulative distribution of cost (for $\rho = 0.2$) for various values of Γ for the robust sorting problem. We see that $\Gamma = 20$ dominates the nominal case $\Gamma = 0$, which in turn dominates $\Gamma = 100$ that appears over conservative. In particular, it is clear that not only the robust solution for $\Gamma = 20$ has lower variability than the nominal solution, it leads to a more favorable distribution of cost.

Γ	$\overline{Z}(\Gamma)$	% Change in $\overline{Z}(\Gamma)$	$\sigma(\Gamma)$	% Change in $\sigma(\Gamma)$
0	8822	0 %	501.0	0.0 %
10	8827	0.056 %	493.1	-1.6 %
20	8923	1.145 %	471.9	-5.8 %
30	9059	2.686 %	454.3	-9.3 %
40	9627	9.125 %	396.3	-20.9 %
50	10049	13.91 %	371.6	-25.8 %
60	10146	15.00 %	365.7	-27.0 %
70	10355	17.38 %	352.9	-29.6 %
80	10619	20.37 %	342.5	-31.6 %
100	10619	20.37 %	340.1	-32.1 %

Table 2: Influence of Γ on $\overline{Z}(\Gamma)$ and $\sigma(\Gamma)$.

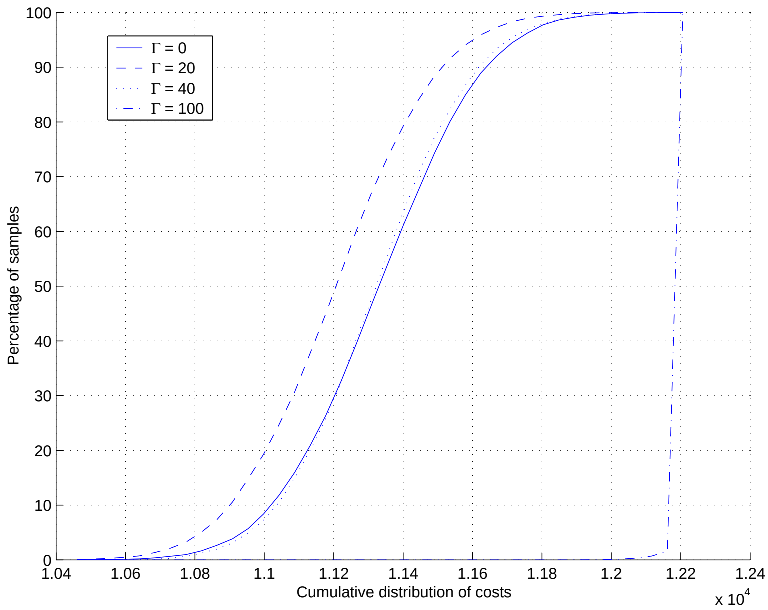

Figure 3: The cumulative distribution of cost (for $\rho = 0.2$) for various values of Γ for the robust sorting problem.

6.3 The Robust Shortest Path Problem

Given a directed graph $G = (\mathcal{N} \cup \{s,t\}, \mathcal{A})$, with non-negative arc cost c_{ij}, $(i,j) \in \mathcal{A}$, the shortest $\{s,t\}$ path problem seeks to find a path of minimum total arc cost from the source node s to the terminal node t. The problem can be modeled as a $0-1$ integer programming problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = \begin{cases} 1, & \text{if } i=s \\ \text{-1}, & \text{if } i=t \\ 0, & \text{otherwise,} \end{cases} \\
& \boldsymbol{x} \in \{0,1\}^{|\mathcal{A}|},
\end{aligned} \tag{31}$$

The shortest path problem surfaces in many important problems and has a wide range of applications from logistics planning to telecommunications [1]. In these applications, the arc costs are estimated and subjected to uncertainty. The robust counterpart is then:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + \max_{\{S|\ S\subseteq\mathcal{A},|S|=\Gamma\}} \sum_{(i,j)\in S} d_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = \begin{cases} 1, & \text{if } i=s \\ \text{-1}, & \text{if } i=t \\ 0, & \text{otherwise,} \end{cases} \\
& \boldsymbol{x} \in \{0,1\}^{|\mathcal{A}|}.
\end{aligned} \tag{32}$$

Using Dijkstra's algorithm [10], the shortest path problem can be solved in $O(|\mathcal{N}|^2)$, while Algorithm A runs in $O(|\mathcal{A}||\mathcal{N}|^2)$. In order to test the performance of Algorithm A, we construct a randomly generated directed graph with $|\mathcal{N}| = 300$ and $|\mathcal{A}| = 1475$ as shown in Figure 4. The starting node, s is at the origin $(0,0)$ and the terminal node t is placed in coordinate $(1,1)$. The nominal arc cost, c_{ij} equals to the euclidean distance between the adjacent nodes $\{i,j\}$ and the arc cost deviation, d_{ij} is set to γc_{ij}, where γ is uniformly distributed in $[0,8]$. Hence, some of the arcs have cost deviations of at most eight times of their nominal values. Using Algorithm A (calling Dijkstra's algorithm $|\mathcal{A}|+1$ times), we solve for the complete set of robust shortest paths (for various Γ's), which are drawn in bold in Figure 4.

We simulate the distribution of the path cost by subjecting the arc cost to random perturbations. In each instance of the simulation, every arc (i,j) has cost that is independently perturbed, with probability ρ, from its nominal value c_{ij} to $c_{ij}+d_{ij}$. Setting $\rho = 0.1$, we generate $20,000$ random scenarios and plot the distributions of the path cost for $\Gamma = 0, 3, 6$ and 10, which are shown in Figure 5. We observe that as Γ increases, the nominal path cost also increases, while cost variability decreases.

In Figure 6 we report the cumulative distribution of cost (for $\rho = 0.1$) for various values of Γ for the robust shortest path problem. Comparing the distributions for $\Gamma = 0$ (the nominal problem) and

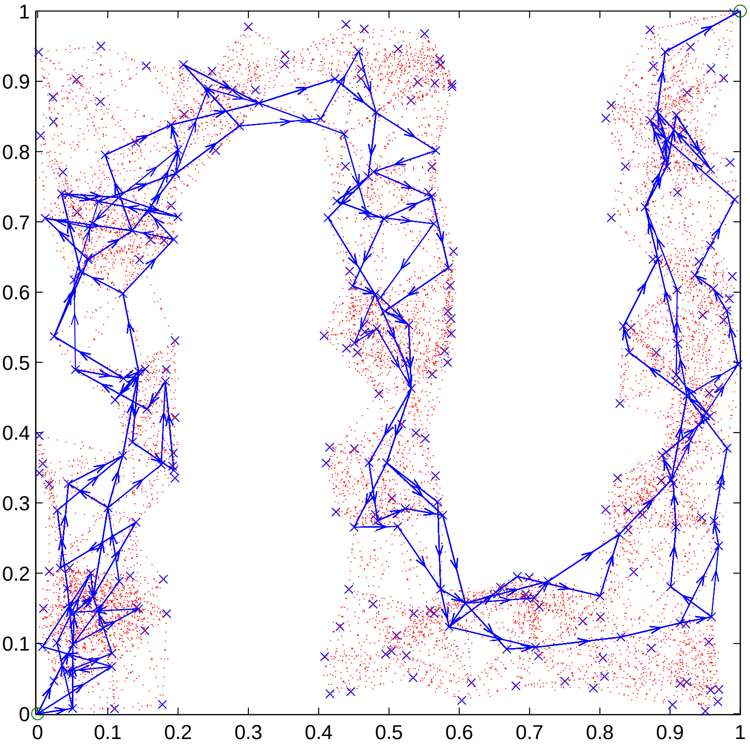

Figure 4: Randomly generated digraph and the set of robust shortest $\{s, t\}$ paths for various Γ values.

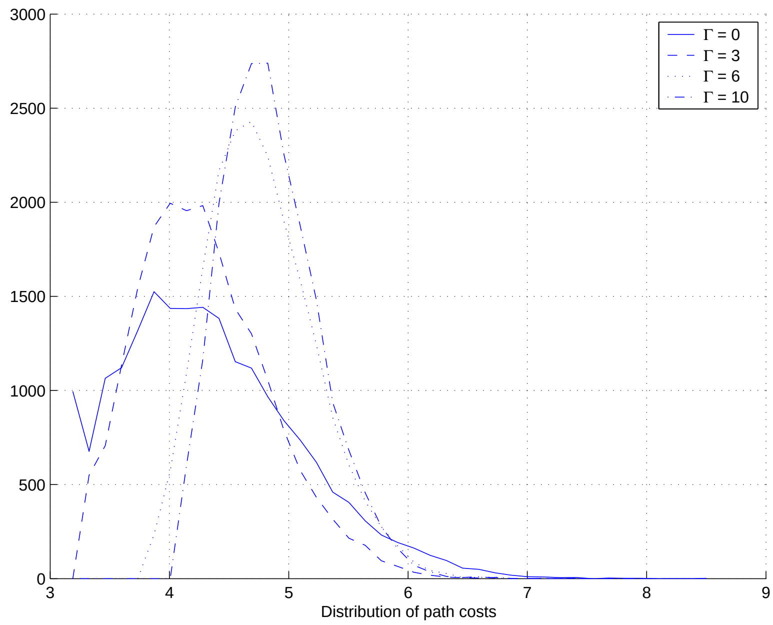

Figure 5: Influence of Γ on the distribution of path cost for $\rho = 0.1$.

$\Gamma = 3$, we can see that none of the two distributions dominate each other. In other words, even if a decision maker is primarily cost conscious, he might still select to use a value of Γ that is different than zero, depending on his risk preference.

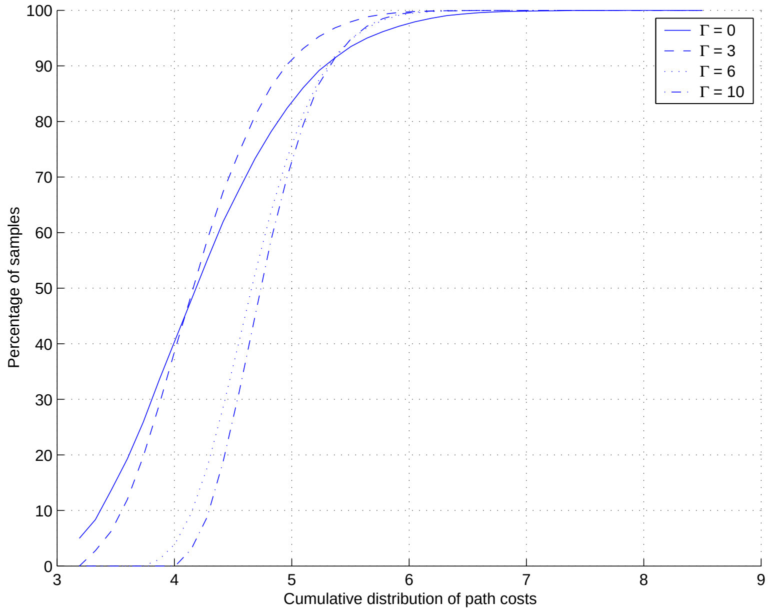

Figure 6: The cumulative distribution of cost (for $\rho = 0.1$) for various values of Γ for the robust shortest path problem.

7 Conclusions

We feel that the proposed approach has the potential of being practically useful especially for combinatorial optimization and network flow problems that are subject to cost uncertainty. Unlike all other approaches that create robust solutions for combinatorial optimization problems, the proposed approach retains the complexity of the nominal problem or its approximability guarantee and offers the modeler the capability to control the tradeoff between cost and robustness by varying a single parameter Γ. For arbitrary discrete optimization problems, the increase in problem size is still moderate, and thus the proposed approach has the potential of being practically useful in this case as well.

References

[1] Ahuja, R., T. Magnanti, J. Orlin. (1993): Network flows: theory, algorithms, and applications. *Prentice Hall*, New York.

[2] Averbakh, I. (2001): On the complexity of a class of combinatorial optimization problems with uncertainty, *Math. Prog. 90*, 263-272.

[3] Ben-Tal, A., Nemirovski, A. (2000): Robust solutions of linear programming problems contaminated with uncertain data, *Math. Prog.* 88, 411-424.

[4] Ben-Tal, A., Nemirovski, A. (1999): Robust solutions to uncertain programs., *Oper. Res. Lett.* 25, 1-13.

[5] Ben-Tal, A., Nemirovski, A. (1998): Robust convex optimization., *Math. Oper. Res.* 23, 769-805.

[6] Ben-Tal, A., El-Ghaoui, L., Nemirovski, A. (2000): Robust semidefinite programming, (in: Saigal, R., Vandenberghe, L., Wolkowicz, H., eds., *Semidefinite programming and applications*, Kluwer Academic Publishers.

[7] Bertsimas, D., M. Sim (2001): The Price of Robustness, *Working Paper*, Operations Research Center, MIT.

[8] Birge, J. R. (1982): The value of the stochastic solutions in stochastic linear programs with fixed recourse. *Math. Prog.* 24, 314-325.

[9] Dantzig, G. B. (1955): Linear programming under uncertainty. *Mgmt. Sci.* 197-206.

[10] Dijkstra, E. W. (1959): A note on two problems in connection with graphs. *Numerische Mathematik*, 269-271.

[11] El-Ghaoui, Lebret, H. (1997): Robust solutions to least-square problems to uncertain data matrices., *SIAM J. Matrix Anal. Appl.* 18, 1035-1064.

[12] El-Ghaoui, L., Oustry, F., Lebret, H. (1998): Robust solutions to uncertain semidefinite programs., *SIAM J. Optim.* 9, 33-52.

[13] Kouvelis, P., G. Yu (1997): Robust discrete optimization and its applications, *Kluwer Academic Publishers*, Norwell, MA.

[14] Mulvey, John M., Robert J. Vanderbei, and Stavros A. Zenios (1995): Robust optimization of large-scale systems, *Oper. Res.* 43, 264-281.

[15] Nemhauser, G., L. Wolsey (1988): Integer and combinatorial optimization. *John Wiley and Sons*, New York.

[16] R. Schultz, L. Stougie and M. van der Vlerk (1998): Solving stochastic programs with complete integer recourse by enumeration: A framework using Grobner basis reductions, *Math. Prog.* 83, 1998, 229-252.

[17] Soyster, A.L. (1973): Convex programming with set-inclusive constraints and applications to inexact linear programming, *Oper. Res.* 21, 1154-1157.

[18] Srinivasan, V., and G. L. Thompson (1972): An operator theory of parametric programming for the transportation problem. *Naval Research Logistics Quarterly* 19, 205-252.