Robust Optimization with Multiple Ranges: Theory and Application to Pharmaceutical Project Selection

Ruken Düüzgün and Aurélie Thiele
Operations Research Scientist, Marriott International, Bethesda, MD, ruken.duzgun@marriott.com
Visiting Associate Professor, MIT, Cambridge, MA, aurelie@mit.edu

Abstract We present a robust optimization approach when the uncertainty in objective coefficients is described using multiple ranges for each coefficient. This setting arises when the value of the uncertain coefficients, such as cash flows, depends on an underlying random variable, such as the effectiveness of a new drug. Traditional robust optimization with a single range per coefficient would require very large ranges in this case and lead to overly conservative results. In our approach, the decision-maker limits the number of coefficients that fall within each range; he can also limit the number of coefficients that deviate from their nominal value in a given range. This is particularly useful for the manager of a pharmaceutical company who aims at constructing a portfolio of R&D projects, or drugs to be developed and tested for commercialization. Modeling multiple ranges requires the use of binary variables in the uncertainty set. We show how to develop tractable reformulations using a concept called total unimodularity and apply our approach to a R&D project selection problem when cash flows are uncertain. Furthermore, we develop a robust ranking heuristic, where the manager ranks the projects according to densities, while incorporating the budgets of uncertainty but without requiring any optimization procedure, to help the manager gain insights into the solution.

Keywords Robust optimization; R&D project selection; total unimodularity; distributional information; multiple ranges; ranking heuristic

1. Introduction

Robust optimization addresses data uncertainty by assuming that uncertain parameters belong to a bounded, convex uncertainty set. It maximizes the minimum value of the objective over that uncertainty set, while ensuring feasibility for the worst-case value of the constraints. This approach was pioneered by Soyster [21] in the 1970s; however, his model required that each uncertain parameter be equal to its worst-case value, and thus was deemed too conservative for practical implementation. In the mid-1990s, Ben-Tal and Nemirovski [2, 3] and El-Ghaoui and Lebret [15] presented tractable mathematical reformulations, based on ellipsoidal uncertainty sets, that turned linear programming problems into second-order cone problems and reduced the conservatism of Soyster’s approach.

Bertsimas and Sim [7, 8] and Bertsimas et al. [9] investigated in the early 2000s the special case where the uncertainty set is a polyhedron. Specifically, the uncertainty set consists of range forecasts (confidence intervals) for each parameter and a constraint called a budget-of-uncertainty constraint, which limits the number of coefficients that can take their worst-case value. The approach preserves the degree of complexity of the problem (the robust
counterpart of a linear problem is linear) and allows the decision-maker to control the degree of conservatism of the solution. Robust optimization remains the focus of substantial research efforts; recent theoretical advances include the development of adjustable optimization \cite{4} and adaptable optimization \cite{6} to incorporate information revealed over time (see Düzgün and Thiele \cite{13} for a review of robust dynamic models), while it has been successfully applied to a variety of areas, as documented in Bertsimas et al. \cite{10}. Ben-Tal et al. \cite{5} provides an extensive treatment of the topic. The reader is further referred to Gabrel et al. \cite{16} for a review of advances in robust optimization over the past five years.

In this paper, we focus on problem setups where the ranges taken by uncertain coefficients depend on the realizations of underlying random variables. This problem arises for instance in R&D project selection, where project cash flows are uncertain but also depend on the effectiveness of the underlying compound tested by the pharmaceutical company. Our high-level goal is to investigate robust optimization in this setting. Project selection requires binary variables, for which ellipsoidal uncertainty sets are ill-suited as they lead to nonlinear integer problems; therefore, we will focus throughout this paper on polyhedral uncertainty sets, specifically, sets with range forecasts and budget-of-uncertainty constraints. Because a drug may fail clinical trials, or emerge as a successful treatment option for most patients, the order of magnitude of the potential cash flows can vary greatly. The traditional robust optimization approach, with a single range for each uncertain coefficient, would thus require very large ranges and lead to overly conservative solutions. The multi-range robust optimization approach we propose allows for a more realistic description of uncertainty. While Metan and Thiele \cite{19} introduces multiple ranges for product demand in a simple two-stage robust revenue management problem for a single product, that approach is an hybrid between robust optimization and stochastic programming, where the decision-maker gains advance knowledge of the range that product demand will fall into. It incorporates neither binary variables nor budgets of uncertainty, and focuses on the impact of scenario probabilities on the quality of the optimal solution.

The Research and Development (R&D) project selection problem has been studied since the 1960s. Competition between R&D companies has increased the importance of funding projects that would best meet their needs. While many methods to identify these projects have been investigated, there is no consensus on their practical effectiveness. Martino \cite{18} summarizes the methods available for selecting R&D projects, which can be classified in four groups: ranking methods, economic models, portfolio or optimization models and ad-hoc methods. Early studies of the R&D project selection problem mostly use ranking methods. The most common ones are scoring models and the analytic hierarchy procedure (AHP) (see Baker and Freeland \cite{1} for a review of these approaches.) Economic methods, which are recommended by Martino \cite{18}, consider the cash flows involved with the project, using metrics such as net present value (NPV), internal rate of return (IRR) and cash flow payback. Portfolio optimization methods implement mathematical programming to find the projects, from a candidate project list, that would give the maximum payoff to the firm. For instance, Childs and Triantis \cite{11} use a real options framework in order to examine dynamic R&D investment policies and valuation of R&D programs. Data envelopment analysis (DEA) is another method for solving R&D project selection decisions. For instance, Eilat et al. \cite{14} use a methodology based on an extended DEA that quantifies some qualitative concepts embedded in the balanced scorecard (BSC) approach. They employ a DEA-BSC model first to evaluate individual R&D projects, and then to evaluate alternative R&D portfolios.

The multi-range approach to robust optimization, which builds upon Düzgün \cite{12}, is particularly well-suited to the context of project portfolio selection for pharmaceutical companies due to the nature of the R&D process. Project selection problems include high levels of uncertainty in future cash flows; however, the most common approaches to project selection replace uncertain parameters by their expected values or rely on traditional, stochastic descriptions of randomness. As mentioned above, the classical robust optimization approach
also suffers from over-conservatism in this setup due to the large ranges that would be required to implement it. This makes multi-range robust optimization a novel theoretical extension of robust optimization with valuable practical applications.

**Contributions.** Our contributions to the literature are as follows.

- We define the multi-range robust optimization framework and derive tractable reformulations.
- In particular, we show that the linear relaxation of the worst-case problem (which computes the worst-case objective for a given strategy and requires binary variables to model multiple ranges) has integer optimal solutions in both robust optimization models we consider.
- We present a robust ranking heuristic to identify projects to fund without any optimization and test it in numerical experiments.

To the best of our knowledge, we are the first to incorporate the idea of robust ranking to a range- and budgets-of-uncertainty-based description of uncertainty.

**Outline.** Section 2 introduces the generic multi-range robust optimization approach. In Section 3, we apply our methodology to a project selection problem. Section 4 introduces the heuristic based on robust ranking. Finally, Section 5 presents the numerical experiments.

### 2. Multi-Range Robust Optimization

We assume here that we have multiple ranges that the uncertain values can take values from. For notational simplicity, we assume that each uncertain parameter has the same number \( m \) of possible ranges, but the approach can be extended easily to the case where the number of ranges depends on the uncertain parameter. We will analyze two cases:

1. The simple case where the (pessimistic) decision-maker assumes that each uncertain parameter takes the worst value of the range it falls into, and the maximum number of parameters that can fall in a given range is bounded by a budget of uncertainty.

2. The more complex case where the decision-maker extends the setup in Case 1 to introduce another family of budgets of uncertainty limiting the number of parameters that can take their worst-case value in a given range. This allows some parameters to be equal to their nominal value, rather than their worst-case value, in that range.

#### 2.1. Case 1: Without a Budget For the Deviations Within The Ranges

Let \( c_i^{-k} \), resp. \( c_i^{+k} \) be the lower, resp. higher, bound of range \( k \) for parameter \( i, i = 1, \ldots, n, k = 1, \ldots, m \). The budget \( \Gamma_k \) constrains the maximum number of coefficients that can fall within range \( k, k = 1, \ldots, m \). (The decision maker can also choose to introduce these budgets only for the lowest ranges, corresponding to the most conservative outcomes, to limit the conservatism of the approach.)

The robust problem can be formulated as a mixed-integer programming problem (MIP):

\[
\begin{align*}
\max \min_{x \in X} c'x \\
\text{s.t. } c_i^{-k} y_i^k & \leq c_i^k \leq c_i^{+k}, \forall i, k, \\
\sum_{k=1}^{m} y_i^k & = 1, \forall i, \\
\sum_{i=1}^{n} y_i^k & \leq \Gamma_k, \forall k, \\
c_i & = \sum_{k=1}^{m} c_i^k, \forall i, \\
y_i^k & \in \{0,1\}, \forall i, k.
\end{align*}
\]
The tractability of the robust optimization paradigm relies on the decision-maker’s ability to convert the inner minimization problem into a maximization problem, of such a structure that the master maximization problem (incorporating the outer maximization problem and the new inner maximization problem) can be solved efficiently. Strong duality has emerged as the tool of choice to implement this conversion [3]; however, the model of uncertainty we propose requires the use of integer (binary) variables, which makes the rewriting of a minimization problem as an equivalent maximization one considerably more difficult. It is thus natural to investigate whether the linear relaxation of the inner minimization problem in Problem (1) yields binary \( y \) variables at optimality. This is the purpose of Lemma 2.1.

**Lemma 2.1.** The linear relaxation of the inner minimization problem:

\[
\min \mathbf{c}' \mathbf{x} \\
\text{s.t. } c_i^k y_i^k - c_i^k y_i^k, \quad \forall i, k, \\
\sum_{i=1}^m y_i^k = 1, \quad \forall i, \quad \sum_{k=1}^n c_i = \sum_{k=1}^m c_i^k, \quad \forall i, \\
\sum_{i=1}^m y_i^k \leq \Gamma_k, \quad \forall k, \\
y_i^k \in \{0, 1\}, \quad \forall i, k.
\]

has a binary optimal vector \( \mathbf{y} \) for any given integer \( \Gamma_i \) and nonnegative vector \( \mathbf{x} \).

**Proof.** The objective is a minimization over \( \mathbf{c} \) of \( \mathbf{c}' \mathbf{x} \) where \( c_i = \sum_{k=1}^m c_i^k \) for all \( i \) and \( \mathbf{x} \) is non-negative. Hence, \( c_i^k \) will take the minimum value in its range, i.e., \( c_i^k = c_i^k - y_i^k \) at optimality for all \( i, k \). It follows that \( c_i = \sum_{k=1}^m c_i^k = \sum_{k=1}^m c_i^k - y_i^k \) for all \( i \) and the feasible set is reduced to \( \sum_{k=1}^m y_i^k = 1, \forall i, \sum_{i=1}^n y_i^k \leq \Gamma_k, \forall k, \) and \( y_i^k \in \{0, 1\}, \forall i, k \). The feasible set of the linear relaxation has binary extreme points, thus proving the lemma.

This allows us to derive a tractable reformulation of Problem (1).

**Theorem 2.2.** Problem (1) has the equivalent robust linear formulation:

\[
\max_{\mathbf{p}, \gamma, \mathbf{z}, \mathbf{x}} \left\{ \sum_{i=1}^n p_i - \sum_{k=1}^m \gamma^k \Gamma_k - \sum_{i=1}^n \sum_{k=1}^m z_i^k \right\} \\
\text{s.t. } p_i - \gamma^k - z_i^k \leq c_i^k - c_i^k x_i, \quad \forall i, k, \\
\mathbf{x} \in \mathcal{X}, \\
\gamma^k, z_i^k \geq 0, \quad \forall i, k.
\]

**Proof.** As in the proof of Lemma 2.1, we notice that, due to the non-negativity of the vector \( \mathbf{x} \), the optimal objective coefficients in the robust optimization framework are always achieved at the low end of the range. Therefore, we can rewrite the group of constraints:

\[
c_i^k - y_i^k \leq c_i^k \leq c_i^k + y_i^k, \quad c_i = \sum_{k=1}^m c_i^k,
\]

as:

\[
c_i = \sum_{k=1}^m c_i^k - y_i^k.
\]

We use Lemma 2.1 to rewrite Problem (2) as:

\[
\min_{\mathbf{y}} \sum_{i=1}^n \sum_{k=1}^m c_i^k y_i^k x_i, \\
\text{s.t. } \sum_{k=1}^m y_i^k = 1, \quad \forall i, \\
\sum_{i=1}^n y_i^k \leq \Gamma_k, \quad \forall k, \\
0 \leq y_i^k \leq 1, \quad \forall i, k.
\]
which is a linear programming problem with a non-empty, bounded feasible set. We can then invoke strong duality to reformulate the minimization as a maximization problem, i.e., replace the primal formulation by its dual. Re-injecting yields Problem (3).

2.2. Case 2: With a Budget For the Deviations Within the Ranges

In practice, it is unlikely that every single uncertain parameter will take the worst-case value of the range it falls in. The purpose of this section is to extend the robust optimization approach presented in Section 2.1 to the case where the manager also decides how many parameters, at most, can take the worst-case value in the ranges they are in.

As before, the uncertain coefficients satisfy:

\[
\begin{align*}
c_i &= \sum_{k=1}^{m} c_i^k, \quad \forall i, \\
|c_i^k - y_i^k| &\leq c_i^k, \quad \forall i, k, \\
\sum_{k=1}^{m} y_i^k &= 1, \quad \forall i, \\
\sum_{i=1}^{n} y_i^k &\leq \Gamma_k, \quad \forall k, \\
y_i^k &\in \{0, 1\}, \quad \forall i, k.
\end{align*}
\]

Because we need to define the deviation of each parameter within its given range, we further assume that the nominal value of parameter \(i\) in range \(k\), denoted \(\bar{c}_i^k\), is known for all \(i = 1, \ldots, n\) and \(k = 1, \ldots, m\). The measure of uncertainty for parameter \(i\) of range \(k\) is then defined as \(z_i^k = c_i^k - \bar{c}_i^k\) for all \(i = 1, \ldots, n\) and \(k = 1, \ldots, m\). Again, because the decision variables are non-negative, the part of the range above the nominal value will not be used in the robust optimization approach and the optimal uncertain coefficients satisfy:

\[
c_i = \sum_{k=1}^{m} (c_i^k - c_i^k z_i^k) y_i^k,
\]

where \(z_i^k\) is the scaled deviation of coefficient \(i\), \(i = 1, \ldots, n\), from its nominal value in range \(k\), \(k = 1, \ldots, m\) with:

\[
\begin{align*}
\sum_{i=1}^{n} \sum_{k=1}^{m} z_i^k &\leq \Gamma, \\
0 &\leq z_i^k \leq 1, \quad \forall i, k.
\end{align*}
\]

**Lemma 2.3.** For any feasible \(x \in X\), the worst-case objective can be computed as a mixed-integer programming problem:

\[
\begin{align*}
\min_{y, u} \sum_{i=1}^{n} \sum_{k=1}^{m} x_i \left( c_i^k y_i^k - c_i^k u_i^k \right) \\
\text{s.t.} \quad u_i^k &\leq y_i^k, & \forall i, k, \hspace{1cm} (5) \\
\sum_{i=1}^{n} \sum_{k=1}^{m} u_i^k &\leq \Gamma, \\
\sum_{k=1}^{m} y_i^k &= 1, & \forall i, \\
\sum_{i=1}^{n} y_i^k &\leq \Gamma_k, & \forall k, \\
y_i^k &\in \{0, 1\}, & \forall i, k, \\
u_i^k &\geq 0, & \forall i, k.
\end{align*}
\]
Proof. Defining $u_i^k = z_i^k y_i^k$, we obtain:

$$c_i^k = c_i^k - c_i^k u_i^k, \forall i, k,$$

where $0 \leq u_i^k \leq y_i^k$. We conclude using that it is suboptimal to have $z_i^k > 0$ when $u_i^k = 0$ for any $i, k$.

The following lemma is key to the tractability of the multi-range robust optimization approach.

**Lemma 2.4.** The constraint matrix of Problem (5) is totally unimodular.

*Proof.* A matrix obtained by row/column operations on a totally unimodular matrix is totally unimodular [20]. The matrix $A$ below is the constraint matrix of Problem (5) where the columns represent the variables $[u \ y]$.

$$C = \begin{bmatrix}
I_{nm} & -I_{nm} \\
1_{1 \times nm} & 0_{1 \times nm} \\
0_{m \times nm} & A_{n \times nm} \\
0_{n \times nm} & B_{m \times nm}
\end{bmatrix}$$

where $I_{nm}$ is the $nm \times nm$ identity matrix and matrix $A_{n \times nm}$ has the following structure:

$$\begin{pmatrix}
1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & \vdots & \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\
0 & \cdots & 0 & \cdots & 1 & \cdots & 1
\end{pmatrix}$$

Specifically, $A$ is defined as:

$$A_{i,j} = \begin{cases} 
1 & \text{if } (i-1)m < j \leq im \\
0 & \text{otherwise},
\end{cases}$$

Matrix $B_{m \times nm}$ has the following structure:

$$\begin{pmatrix}
1 & \cdots & 0 & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & 1 & 0 & \cdots & 1 & \cdots & 0 & \cdots & 1
\end{pmatrix}$$

Specifically, $B$ has the following structure:

$$B = (I_{m \times m} \ I_{m \times m} \ \cdots \ I_{m \times m})$$

We will do the following operations on $C$.

1) Let $R_j$ be the $j^{th}$ row and $R^j$ the $j^{th}$ column of $C$. By doing the row operations, we obtain the $r$-th version of the matrix $C$ which is denoted by $(C)^r$.

For $j = nm + 2$ to $nm + 2 + m$, do

$$-R_j + R_{nm+1} \rightarrow R_{nm+1}$$

and call the resulting matrix $(C)^m$.

Now, for $j = 1$ to $nm$, do

$$R^j + R^{j+nm} \rightarrow R^{j+nm}.$$ 

and call the resulting matrix $(C)^m(n+1)$. 

$$\begin{array}{l}
\begin{bmatrix}
I_{nm} & -I_{nm} \\
1_{1 \times nm} & 0_{1 \times nm} \\
0_{m \times nm} & A_{n \times nm} \\
0_{n \times nm} & B_{m \times nm}
\end{bmatrix}
\end{array}$$
2) At the end of these row/column operations we obtain the matrix \((C)^{m(n+1)}\), which is
\[
(C)^{m(n+1)} = \begin{pmatrix}
I_{nm} & 0_{nm} \\
1_{1 \times nm} & 0_{1 \times nm} \\
0_{m \times nm} & A_{n \times nm} \\
0_{n \times nm} & B_{m \times nm}
\end{pmatrix}
\]

To conclude the proof, we will need the following result.

Lemma 2.5 (Nemhauser and Wolsey [20] p. 544). Let \(A\) be a \((0,1,-1)\) matrix with no more than two nonzero elements in each column. Then \(A\) is totally unimodular if and only if the rows of \(A\) can be partitioned into two subsets \(Q_1\) and \(Q_2\) such that if a column contains two nonzero elements, the following statements are true:

a. If both nonzero elements have the same sign, then one is in a row contained in \(Q_1\) and the other is in a row contained in \(Q_2\).

b. If the two nonzero elements have opposite sign, then both are in rows contained in the same subset.

The matrix \((C)^{m(n+1)}\) satisfies these conditions of total unimodularity. Since a matrix obtained by row/column operations on a totally unimodular matrix is also totally unimodular, our constraint matrix \(C\) is totally unimodular.

Theorem 2.6. The robust counterpart is equivalent to the following problem, with a linear objective and linear constraints added to the original feasible set:

\[
\begin{align*}
\max_{\pi, \gamma, \gamma_0, x, z} & \left\{ \sum_{i=1}^{n} p_i - \sum_{i=1}^{n} \sum_{k=1}^{m} z_i^k - \sum_{k=1}^{m} \gamma^k \Gamma_k - \Gamma \gamma_0 \right\} \\
\text{s.t.} & \quad \pi^k_i + \gamma_0 \geq \bar{c}^k_i x_i, \quad \forall i, k, \\
& \quad \pi^k_i + p_i - \gamma^k - z_i^k \leq \bar{c}^k_i x_i, \quad \forall i, k, \\
& \quad x \in X \\
& \quad \gamma^k, \gamma_0, \pi^k_i, z_i^k \geq 0 \quad \forall i, k.
\end{align*}
\]

Proof. Since the constraint matrix of Problem \([5]\) is totally unimodular (Lemma 2.4) and the right-hand-side values of the constraints are integer, the linear relaxation of the problem has integer optimal solutions. It follows from strong duality, because the feasible set of the linear relaxation of Problem \([5]\) is non-empty and bounded, that Problem \([5]\) and the dual of its linear relaxation have the same optimal objective. Reinjecting the dual yields Problem \([6]\). \(\square\)

3. Application to Pharmaceutical Project Selection

3.1. Problem Setup

We now apply the setting described in Section 2 to a pharmaceutical project selection problem where cost and net present values (NPV) of the drugs being considered for development are uncertain. The manager must decide in which projects to invest over a finite time horizon. Each project has known cash requirements at each stage of its development (for notational simplicity, we assume all projects have the same number of stages; this corresponds for instance to the case of drug trials of small, medium and large scale leading to possible approval by the Food and Drug Administration in the United States), but cash flows during and at the end of development are uncertain and depend on underlying random variables, such as the effectiveness of the active compounds or the market response to the new product. These random variables are realized only once (e.g., the drug compound is effective for the disease being treated), so that the coefficients for a given project all fall in the low range or all fall in the high range. We allow for cash flows to be generated during development as the company might file for patents or generate monetary value from the
results of the intermediary stages; the biggest cash flows, however, will be generated at the end of the development phase.

We assume that there are two uncertainty ranges for each cash flow: a project might be successful and has high cash flows, or it might be a failure and has low cash flows. Note that cash flows are non-zero, even in the low state, as the drug might be found to be effective on a subset of the patients and retain some market value. Because no new information is revealed during the time horizon in this robust optimization setting, we do not consider the possibility of stopping a project before the end of the development phase.

The goal is to maximize the worst-case cumulative Net Present Value of the projects the manager invests in, where the worst case is computed over the uncertainty sets described in Sections 2.1 and 2.2 subject to constraints on the amount of money available at each time period to spend on development. We will use the following notation throughout the paper.

**General and cost parameters**

- \( n \): number of projects,
- \( T \): number of time periods,
- \( S \): number of development phases for each project,
- \( B_t \): available budget for the time period \( t \) where \( t = 1, \ldots, T \),
- \( CD_{i,s} \): development cost of project \( i \) in phase \( s \),
- \( r \): discount rate at each time period.

**Cash flow parameters**

- \( CF_{l,s}^i \): lower bound of cash flow of project \( i \) in phase \( s \) if the project is unsuccessful,
- \( CF_{h,s}^i \): nominal value of the cash flow of project \( i \) in phase \( s \) if the project is unsuccessful,
- \( CF_{h,s}^i \): upper bound of cash flow of project \( i \) in phase \( s \) if the project is unsuccessful,
- \( \hat{CF}_{l,s}^i \): measure of uncertainty for cash flow of project \( i \) in phase \( s \) in low range \((= CF_{l,s}^i - CF_{l,s}^i)\),
- \( CF_{l,s}^i \): lower bound of cash flow of project \( i \) in phase \( s \) if the project is successful,
- \( CF_{h,s}^i \): nominal value of the cash flow of project \( i \) in phase \( s \) if the project is successful,
- \( CF_{h,s}^i \): upper bound of cash flow of project \( i \) in phase \( s \) if the project is successful,
- \( \hat{CF}_{h,s}^i \): measure of uncertainty for cash flow of project \( i \) in phase \( s \) in high range \((= CF_{h,s}^i - CF_{h,s}^i)\).

**Robust optimization parameters and decision variables**

- \( \Gamma_l \): uncertainty budget that restricts the number of projects whose cash flows will be in the low range,
- \( \Gamma \): uncertainty budget that restricts the number of projects whose cash flows deviate from their nominal value within their given range,
- \( x_{i,\tau} \): 1 if the project \( i \) is selected to begin at time \( \tau \), 0 otherwise,
- \( y_i \): 1 if the project \( i \) is in its low range (unsuccessful), 0 otherwise.

The deterministic project selection problem is formulated as (each project can be selected at most once):

\[
\max \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} x_{i,\tau} \left[ \frac{\sum_{s=1}^{S} CF_{i,s}}{(1+r)^{\tau-1}} \right] \\
\text{s.t.} \sum_{i=1}^{n} \sum_{\tau=\max\{1, T-S+1\}}^{T} CD_{i,t-\tau+1} x_{i,\tau} \leq B_t \quad \forall t, \\
\sum_{\tau=1}^{T} x_{i,\tau} \leq 1, \quad \forall i, \\
x_{i,\tau} \in \{0,1\}, \quad \forall i, \tau.
\]
3.2. Case 1: Robust Optimization Without a Budget for the Deviation Within the Ranges

First, we consider the simple case where the manager only limits the number of projects that will be unsuccessful, and assumes that each cash flow will take its worst case within a given range. The worst-case cash flows are computed as, for any feasible solution:

\[
\min_{CF_{i,s},y_i} \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau-1}} \left[ \sum_{s=1}^{S} CF_{i,s} \right]
\]

[Total cash flow over time]

s.t.

\[
CF_{i,s} \leq CF_{i,s} \leq CF_{i,s} y_i \quad \forall (i, s)
\]

[Cash flow interval if in low range]

\[
CF_{i,s} (1 - y_i) \leq CF_{i,s} \leq CF_{i,s} (1 - y_i) \forall (i, s)
\]

[Cash flow interval if in high range]

\[
CF_{i,s} + CF_{i,s} = CF_{i,s} \quad \forall (i, s)
\]

[Cash flow is either high or low]

\[
\sum_{i=1}^{n} y_i \leq \Gamma_l
\]

[At most \(\Gamma_l\) projects in low range]

\[
y_i \in \{0, 1\}, \quad \forall i
\]

\[
CF_{i,s}, CF_{i,s}, CF_{i,s} \geq 0 \quad \forall (i, s).
\]

(8)

**Theorem 3.1.** The robust optimization problem in the case without deviation is equivalent to the mixed-integer programming problem:

\[
\max_{x,z} \left\{ \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \sum_{s=1}^{S} \frac{x_{i,\tau}}{(1+r)^{\tau+s-1}} CF_{i,s} - z_i \Gamma_l - \sum_{i=1}^{n} z_i \right\}
\]

subject to

\[
\sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \sum_{t=1}^{T-S+1} CD_{i,t-\tau+1} x_{i,\tau} \leq B_t, \quad \forall t
\]

\[
\sum_{i=1}^{n} x_{i,\tau} \leq 1, \quad \forall i,
\]

\[
z_i + z_i \geq \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau+s-1}} \sum_{s=1}^{S} CF_{i,s} - CF_{i,s}, \quad \forall i,
\]

\[
x_{i,\tau} \in \{0, 1\}, \quad \forall i, \tau,
\]

\[
z_i, z_i \geq 0, \quad \forall i.
\]

(9)

**Proof.** This is a direct application of Model (3) where we have \(CF_{i,s}\) instead of \(c_i\) and we have two uncertainty ranges, \(h\) and \(l\). The proof is the same as for Theorem 2.2. \(\square\)

3.3. Case 2: Robust Optimization With a Budget for the Deviation Within the Ranges

Assume that cash flows for project \(i\) in phase \(s\), with \(i = 1, \ldots, n, s = 1, \ldots, S\), are either in \([CF_{i,s} - \hat{CF}_{i,s}, CF_{i,s} + \hat{CF}_{i,s}]\) or \([CF_{i,s} - \hat{CF}_{i,s}, CF_{i,s} + \hat{CF}_{i,s} + \hat{CF}_{i,s}]\). In line with the framework in Section 2.2, they can be written in mathematical terms as:

\[
CF_{i,s} = (CF_{i,s} - \hat{CF}_{i,s} y_i^l) + (CF_{i,s} - \hat{CF}_{i,s} y_i^h) + \hat{CF}_{i,s},
\]

with \(0 \leq y_i^l, y_i^h \leq 1 \forall i, s\) and \(y_i^l \in \{0, 1\} \forall j \in \{l, h\}\). Since the coefficients must belong to one of the two ranges, we only introduce a budget of uncertainty on the number of coefficients that fall into their low range.
Given feasible binary variables $x_{i,\tau}$, the worst-case cash flows are given by:

$$
\min \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} x_{i,\tau} \left[ \sum_{s=1}^{S} \frac{\text{CF}^l_{i,s} y^l_{i,s} - \text{CF}^l_{i,s} y^h_{i,s} + \text{CF}^h_{i,s} y^l_{i,s} - \text{CF}^h_{i,s} y^h_{i,s}}{(1+r)^{r+\tau-1}} \right] \\
\text{s.t. } u^l_{i,s} \leq y^l_{i,s}, \quad u^h_{i,s} \leq y^h_{i,s}, \quad \forall i, s, \\
\text{s.t. } y^l_{i} + y^h_{i} = 1, \quad \forall i, \\
\sum_{i=1}^{n} y^l_{i} \leq \Gamma_l, \\
\sum_{i=1}^{n} \sum_{s=1}^{S} (u^l_{i,s} + u^h_{i,s}) \leq \Gamma, \\
y^j_{i} \in \{0, 1\}, \quad \forall i, \forall j \in \{l, h\}, \\
u^l_{i,s}, u^h_{i,s} \geq 0, \quad \forall i, s.
$$

**Theorem 3.2.** The robust optimization problem in the case with a deviation budget is given by the mixed-integer programming problem:

$$
\max \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} CD_{i,\tau} x_{i,\tau} \leq B_t, \quad \forall t, \\
\text{s.t. } \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} x_{i,\tau} \leq 1, \quad \forall i, \\
\sum_{s=1}^{S} \pi^l_{i,s} + p_i - \gamma_l - z^l_i \leq \sum_{\tau=1}^{T-S+1} x_{i,\tau} \sum_{s=1}^{S} \frac{\text{CF}^l_{i,s}}{(1+r)^{r+\tau-1}}, \quad \forall i, \\
\sum_{s=1}^{S} \pi^h_{i,s} + p_i - z^h_i \leq \sum_{\tau=1}^{T-S+1} x_{i,\tau} \sum_{s=1}^{S} \frac{\text{CF}^h_{i,s}}{(1+r)^{r+\tau-1}}, \quad \forall i, \\
x_{i,\tau} \in \{0, 1\}, \quad \forall i, \tau, \\
\pi^l_{i,s}, \pi^h_{i,s} \geq 0, \quad \forall i, s, \\
z_i^l, z_i^h \geq 0, \quad \forall i, \\
\gamma_l, \gamma_0 \geq 0.
$$

**Proof.** This is a direct application of Model [9] where we have $\text{CF}_{i,s}$ instead of $c_i$ and we have two uncertainty ranges, $h$ and $l$. The proof is the same as for Theorem 2.6. □

The feasible set can be decomposed as follows:

- The first two groups of constraints are the same as in the deterministic model, representing the maximum amount of money to be allocated at each time period and the fact that a project can be started at most once.
- The third and fourth group of constraints are the dual constraints corresponding to the primary variables $y^l_{i}$ and $y^h_{i}$, respectively, and incorporate the information about the
nominal values of the cash flows. Because one of these decision variables (either $y^l_i$ or $y^h_i$) will be non-zero for each $i$ at optimality, by complementarity slackness, one of the dual constraints will be tight for each $i$, thus determining $p_i$ as a function of the nominal cash flow for that range and the other dual variables. This will bring the nominal cash flows back into the objective.

- The fifth and sixth group of constraints are the dual constraints corresponding to the primary variables $u^l_i$ and $u^h_i$, respectively, and incorporate the information about the uncertainty on the cash flows in each range. At most one of these decision variables (either $u^l_i$ or $u^h_i$) will be non-zero for each $i$ at optimality; if it is non-zero, by complementarity slackness, one of the dual constraints will be tight for each $i$, thus determining either $\pi^l_i$ or $\pi^h_i$ as a function of the uncertainty in that range and the other dual variables. (Otherwise the $\pi^l_i$ and $\pi^h_i$ variables will be at zero.) This will bring the cash flow uncertainty, through the half-range of the confidence intervals, into the objective when needed.

- The other constraints are sign constraints or binary constraints.

4. Robust Ranking Heuristic

While Problem (11) provides an exact formulation of the robust optimization problem, we focus in this section on developing optimization-free heuristics, which would give practitioners more insights into the strategy they implement and the impact of the cash flow parameters. We are motivated by the fact that, when there is only one time period, the project selection problem has the structure of a knapsack problem, for which a well-known heuristic is to rank items by decreasing order of density (value to weight ratio) and fill the knapsack until the next item in the list does not fit (see, for instance, Kellerer et al. [17]). In particular, we provide a robust ranking procedure to rank the projects with uncertain cash flows; to the best of our knowledge, we are the first to present such a ranking procedure in the context of robust optimization.

We will investigate ranking the projects according to decreasing density; as we are motivated by its popularity to solve the generic knapsack problem. Once projects are ranked, we apply the greedy multiple-knapsack heuristic described in Kellerer et al. [17] to generate a candidate solution.

Overview of heuristic.

In the setting with two uncertainty ranges, the cumulative cash flow of a project can take four possible values: low value of the low range, nominal value of the low range, low value of the high range, nominal value of the high range. The high-level idea is to (i) compute four rankings, one for each of the possible metrics, (ii) use the “low value of the low range” ranking until one of the two budgets of uncertainty has been used up, (iii) use either the “nominal value of the low range” ranking or the “low value of the high range” ranking (depending on which budget is not yet zero) until the other budget of uncertainty has been used up, and (iv) complete the procedure using the “nominal value of the high range” ranking.

Ranking procedure.

The heuristic in the simpler case without a budget can be derived by inserting $\Gamma = n$ in the procedure (so that only the parameters $a^l_i$ and $a^h_i$, defined below, are required).

**Step 1** Compute the four following parameters for all projects $i$.

\[
A^h_i = \frac{CF^h_i}{(1+r)CD_i}, \quad A^l_i = \frac{CF^l_i}{(1+r)CD_i}, \quad a^h_i = \frac{CF^h_i}{(1+r)CD_i} - CF^l_i, \quad a^l_i = \frac{CF^l_i}{(1+r)CD_i}.
\]

which are, respectively: the nominal value in the high range (most optimistic outcome of the four), the nominal value in the low range, the low value in the high range and the low value in the low range (most pessimistic outcome of the four). We then rank the projects in decreasing order of the parameters $A^h_i$, $A^l_i$, $a^h_i$, and $a^l_i$, leading to four rankings.
Step 2 Choose the projects corresponding to the biggest \( \min(\Gamma_i, \Gamma) \) in the ranking based on the \( a_i^l \) parameters.

Step 3 If \( \Gamma_i > \Gamma \) (all cash flows will now take their nominal value as we have used up the \( \Gamma \) budget, but the decision maker still expects \( \Gamma_i - \Gamma \) projects to have cash flows in the low range), add \( \Gamma_i - \Gamma \) projects to the ranked list by using the ranking based on the \( A_i^l \) parameters, skipping the projects that have already been selected in Step 2.

Step 4 If \( \Gamma - \Gamma_i > 0 \), add \( \Gamma - \Gamma_i \) projects to the ranked list by using the ranking based on the \( a_i^h \) parameters, skipping the projects that have already been selected in Steps 2 and 3.

Step 5 Continue until all projects are ranked by using the ranking based on the \( A_i^h \) parameters, skipping the projects that have already been selected in Steps 2, 3 and 4.

5. Numerical Example

In this section, we investigate the practical performance of our robust optimization model and heuristic on an example. We focus on the case where we only have one time period, for which the mathematical formulation without uncertainty becomes a well-known knapsack problem. Our goal is to determine whether the robust optimization framework does protect against downside risk as stated, and study the performance of the heuristic compared to the optimal solution.

We tested our formulations and heuristics for 4 data sets. DataSet1 and DataSet2 have 10 projects while DataSet3 and DataSet4 have 20 projects. In all cases, development costs \( (CD_i) \) were generated using a Uniform distribution in \([80 - 120]\), nominal values of low cash flows \( (CF_i^l) \) were generated using Uniform distribution in \((0.5 - 2.5) \cdot CD_i\), and nominal values of high cash flows \( (CF_i^h) \) generated using Uniform distribution in \((2 - 3.5) \cdot CD_i\). For all \( i \), the deviation parameters \( \hat{CF}_i^l, \hat{CF}_i^h \) were selected as \( 0.2 \cdot CF_i^l, 0.2 \cdot CF_i^h \) respectively. The budget for development costs was set to 500 in all cases. In addition, DataSet3 and DataSet4 were also solved for a value of the budget equal to 1,000. The same distributions were used to compute the actual objective using random cash flows once the optimization problem had been solved. The probability of the cash flows being in the low range was taken equal to 0.5.

Optimal solution.

We solved Problem (11) for each data set and for each \( (\Gamma, \Gamma_i) \) combination. Figure 1 shows the histogram of revenues for Data Set 1 and the deterministic model, where parameter values are taken equal to their expected values, here \( (CF_i^h + CF_i^l)/2 \) for all \( i \) (line with square markers) as well as two robust models: \( (\Gamma, \Gamma_i) = (3, 4) \) and \( (\Gamma, \Gamma_i) = (2, 1) \) (lines with lozenge and triangle markers, respectively). These budgets were chosen to have \( \Gamma > \Gamma_i \) in one case and \( \Gamma < \Gamma_i \) in the other. This histogram was generated using 1,000 scenarios. Figure 1.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{revenue_histogram.png}
\caption{Histogram of Revenues. Results for Data Set 1 using 1,000 scenarios.}
\end{figure}
suggests that robust optimization is more conservative than its nominal counterpart (limits upside potential) but decreases the downside risk.

Figures 2 and 3 show the number of iterations versus the budget of uncertainty $\Gamma_l$ for five different $\Gamma$ values, for DataSet1 and DataSet3, respectively. Recall that DataSet1 has 10 projects and DataSet3 has 20. (Our observations remain valid for other values of $\Gamma$, but the corresponding graphs were omitted for graph readability.) We observe that, for each $\Gamma$ value, the number of iterations in the robust optimization models does not differ substantially from the number of iterations in the model when $\Gamma_l$ is close to its bounds ($\Gamma_l = 0$ or $\Gamma_l = 10$), which means that most projects are in the same uncertainty range.) When projects are more evenly assigned to low and high ranges (middle values of $\Gamma_l$), the number of iterations increases, sometimes substantially (see Figure 3, where the top curve corresponds to $\Gamma = 10$).

Since robust optimization maximizes the worst-case cash flow over the uncertainty set, it is natural to evaluate how well it protects against downside risk. Thus, we compute the first and fifth percentile of the distribution of the random objective where we have injected the optimal solution, for DataSet1 and all $(\Gamma, \Gamma_l)$ combinations, using 1,000 scenarios. These results are shown in Tables 1 and 2, respectively. Table 3 shows the expected value of the objective for reference. We see that robust optimization does indeed protect against downside risk, as evidenced in the increase in the values for the first and fifth percentile, with at worst modest performance degradation (decrease in average objective value). For instance, the first and fifth percentiles and the expected revenue for $\Gamma = \Gamma_l = 4$ are 919.5, 981 and 1148.1, respectively, to be compared with their counterparts in the nominal case: 811.5, 879.5 and 1156.3. This represents an increase in downside risk protection (measured by the increase in first or fifth percentile) of 13.3% and 11.5%, respectively, while the expected revenue decreases by only 0.7%.
<table>
<thead>
<tr>
<th>( \Gamma \Gamma_l \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5–10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>811.5</td>
<td>811.5</td>
<td>908.5</td>
<td>919.5</td>
<td>919</td>
<td>919</td>
</tr>
<tr>
<td>1</td>
<td>811.5</td>
<td>908.5</td>
<td>908.5</td>
<td>908.5</td>
<td>919</td>
<td>919</td>
</tr>
<tr>
<td>2</td>
<td>842.5</td>
<td>908.5</td>
<td>908.5</td>
<td>908.5</td>
<td>919</td>
<td>919</td>
</tr>
<tr>
<td>3</td>
<td>842.5</td>
<td>908.5</td>
<td>908.5</td>
<td>908.5</td>
<td>919.5</td>
<td>919</td>
</tr>
<tr>
<td>4</td>
<td>842.5</td>
<td>908.5</td>
<td>908.5</td>
<td>908.5</td>
<td>919.5</td>
<td>919</td>
</tr>
<tr>
<td>5–10</td>
<td>811.5</td>
<td>811.5</td>
<td>908.5</td>
<td>919.5</td>
<td>919</td>
<td>919</td>
</tr>
</tbody>
</table>

Table 1. First percentile values for each \((\Gamma, \Gamma_l)\) pair with DataSet1.

<table>
<thead>
<tr>
<th>( \Gamma \Gamma_l \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5–10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>879.5</td>
<td>879.5</td>
<td>965</td>
<td>981</td>
<td>968</td>
<td>968</td>
</tr>
<tr>
<td>1</td>
<td>879.5</td>
<td>965</td>
<td>965</td>
<td>965</td>
<td>968</td>
<td>968</td>
</tr>
<tr>
<td>2</td>
<td>911</td>
<td>965</td>
<td>965</td>
<td>965</td>
<td>968</td>
<td>968</td>
</tr>
<tr>
<td>3</td>
<td>911</td>
<td>965</td>
<td>965</td>
<td>965</td>
<td>981</td>
<td>968</td>
</tr>
<tr>
<td>4</td>
<td>911</td>
<td>965</td>
<td>965</td>
<td>965</td>
<td>981</td>
<td>968</td>
</tr>
<tr>
<td>5–10</td>
<td>879.5</td>
<td>879.5</td>
<td>965</td>
<td>981</td>
<td>968</td>
<td>968</td>
</tr>
</tbody>
</table>

Table 2. Fifth percentile values for each \((\Gamma, \Gamma_l)\) pair with DataSet1.

<table>
<thead>
<tr>
<th>( \Gamma \Gamma_l \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5–10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1156.3</td>
<td>1156.3</td>
<td>1162.0</td>
<td>1148.1</td>
<td>1117.4</td>
<td>1117.4</td>
</tr>
<tr>
<td>1</td>
<td>1156.3</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1117.4</td>
<td>1117.4</td>
</tr>
<tr>
<td>2</td>
<td>1176.7</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1117.4</td>
<td>1117.4</td>
</tr>
<tr>
<td>3</td>
<td>1176.7</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1148.1</td>
<td>1117.4</td>
</tr>
<tr>
<td>4</td>
<td>1176.7</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1162.0</td>
<td>1148.1</td>
<td>1117.4</td>
</tr>
<tr>
<td>5–10</td>
<td>1156.3</td>
<td>1156.3</td>
<td>1162.0</td>
<td>1148.1</td>
<td>1117.4</td>
<td>1117.4</td>
</tr>
</tbody>
</table>

Table 3. Expected revenue for each \((\Gamma, \Gamma_l)\) pair with DataSet1.

Note that the optimal solution will not change once \(\Gamma\) or \(\Gamma_l\) increases past the number of projects being funded, say \(N\). If \(p\) is the (estimated) probability of project cash flows falling in the low range, a decision-maker interested in protecting his cumulative cash flow against adverse events will select \(\Gamma_l \geq pN\); however, \(N\) cannot be determined before the robust optimization problem has been solved (and depends somewhat on \(\Gamma\) and \(\Gamma_l\), although the dependence is minimal in our experiments: the manager invests in 4 or 5 projects out of 10 in Data Sets 1 and 2, and 10 or 11 projects out of 20 in Data Sets 3 and 4).

Therefore, we recommend that the decision-maker compute Tables 1, 2 and 3 for his own project selection problem, and choose an appropriate \((\Gamma, \Gamma_l)\) pair based on the trade-off between downside risk (measured either by first or fifth percentile) and performance (measured by average objective) that he wishes to achieve. Also note that several \((\Gamma, \Gamma_l)\) pairs have the same optimal solution, due to the use of binary variables, and that what the manager ultimately needs to determine is the strategy he will implement, rather than a specific \((\Gamma, \Gamma_l)\) pair, which would only be used to compute the corresponding optimal strategy anyway. In the case of Data Set 1, we recommend to invest in projects 1, 3, 6, 8, 10; this strategy is optimal for \((\Gamma, \Gamma_l)\) pairs \((3, 4), (4, 4), (0, 3)\) and \((\Gamma, 3)\) for any \(\Gamma \geq 5\). This choice maximizes both first and fifth percentiles over all possible \((\Gamma, \Gamma_l)\) combinations, achieving the biggest shift of the cumulative cash flow distribution to the right.

**Heuristics.**

Table 4 compares the objective function values of the heuristic with the optimal objective function value. For this comparison, the development budget was taken equal to 500 in all
data sets. % Obj. Dif. is the average percent error of heuristics in objective value. # Opt. indicates the number of times the heuristics gives the same objective function value as the optimal solution when all possible \((\Gamma, \Gamma_l)\) pairs are enumerated. We observe that the heuristic can terminate at the true optimal solution for some data sets. Interestingly, the data set for which it performs best in that respect (DataSet2) is also the one where the optimality gap is largest (about 7%). This suggests that in this case, when the heuristic does not terminate at the optimal solution, the optimality gap can be somewhat large. In contrast, the heuristic never terminates at the optimal solution in DataSet3 but the optimality gap is the smallest of the four data sets, at about 2%.

\[
\begin{array}{c|c|c}
\text{DataSet1} & 3.28 & 15/121 \\
\text{DataSet2} & 6.32 & 77/121 \\
\text{DataSet3} & 1.82 & 0/441 \\
\text{DataSet4} & 5.23 & 21/441 \\
\end{array}
\]

Table 4. Comparison between heuristic and optimal solutions.

We now evaluate the optimal solution and the heuristic solutions. We generated 100 scenarios for the cash flows when the probability of falling into the low range is 0.5. We implemented the optimal and heuristic solutions of each data set for these scenarios (again, with a budget of 500 in all cases) and computed the mean and standard deviation of the objective (cumulative discounted cash flows). Table 5 shows the average percentage (absolute) difference in mean and standard deviation of the simulation results over all \((\Gamma, \Gamma_l)\) combinations. We see that using the heuristics rather than the optimal solution does not significantly change the objective average, but does change the standard deviation more significantly. There was no sign pattern in the mean difference or standard deviation difference, which is why we only show absolute values.

\[
\begin{array}{c|c|c}
\text{Heuristic vs Optimal} & \% \text{ Dif.} & \# \text{ Opt.} \\
\text{DataSet1} & 3.28 & 15/121 \\
\text{DataSet2} & 6.32 & 77/121 \\
\text{DataSet3} & 1.82 & 0/441 \\
\text{DataSet4} & 5.23 & 21/441 \\
\end{array}
\]

Table 5. Absolute percentage difference in mean and standard dev. of simulation results over all \((\Gamma, \Gamma_l)\) combinations.

6. Conclusions

We have presented an approach to robust optimization with multiple ranges for each uncertain coefficient. Our proposed multi-range approach gives more flexibility to the decision-maker to specify how many coefficients can fall in each of the ranges and thus allows for a finer description of uncertainty within the robust optimization framework. We derived tractable exact reformulations that do not require any binary variables using total unimodularity. This method can be applied to any uncertainty problem as it does not require any assumption but only that the variable multiplied with the uncertain parameter be nonnegative. We have also provided a robust ranking heuristic according to project densities to provide the R&D project manager with insightful, high-quality solutions without resorting to optimization.
References


