Robust binary linear programming under implementation uncertainty

Jose Ramirez-Calderon, V. Jorge Leon, and Barry Lawrence

ABSTRACT
This article studies binary linear programming problems in the presence of uncertainties that may prevent implementing the computed solution. This type of uncertainty, called implementation uncertainty, is modelled affecting the decision variables rather than model parameters. The binary nature of the decision variables invalidates using existing robust models for implementation uncertainty. The robust solutions obtained are optimal for a worst-case min–max objective. Structural properties allow the reformulation of the problem as a binary linear program. Constraint relaxation and cardinality-constrained parameters control the degree of solution conservatism. An optimization problem permits the selection of solutions from the obtained set of robust solutions. Results from a case study in the context of the knapsack problem suggest the methodology yields solutions that perform well in terms of objective value and feasibility. Furthermore, the selection approach can identify robust solutions with desirable implementation characteristics.

1. Introduction

1.1. Introduction
This article studies binary linear programming problems (BLPs) in the presence of uncertainties that may cause solution values to change during implementation. This type of uncertainty, called implementation uncertainty, is modelled explicitly affecting the decision variables rather than model parameters. Implementation uncertainty may result in implemented solutions that are different from what is prescribed by the BLP. The impact of implementation uncertainty on binary variables acts as if the variable is switching its prescribed value at the time of the implementation; therefore, a solution different from the prescribed one is implemented. Implementation uncertainty inevitably occurs owing to inherent fidelity limitations of problem formulations and unexpected future events, including those caused by exogenous factors such as political directives, regulatory issues or sudden extreme events. Model fidelity limitations are unavoidable in practice owing to restricted time availability during modelling, limited knowledge about the problem at hand, and simplifying model assumptions. Implementing a different solution rather than the prescribed one may cause the objective value to become negatively impacted, leading to a suboptimal value; additionally, the implemented solution may no longer be feasible. This type of uncertainty affecting binary variables hinders applying most of the existing uncertainty models proposed in the related literature. Assuming that only a subset of variables is affected by implementation uncertainty and the others are deterministic, solving a BLP under
1.2. Illustrative example

An investor needs to decide in which of 10 projects to invest. Each project \( i \) possesses an associated profit \( c_i \) and cost \( a_i \). The selection has to be such that it maximizes the profit while maintaining the cost within a budget of \( b = 26 \). Table 1 presents the remaining data for this example.

This problem is an application of the knapsack problem, a well-known binary linear programming problem. The optimal solution for this deterministic version of the problem is a profit of 41 and a total budget requirement of 26.

To introduce the concept of implementation uncertainty, consider that projects 1 and 2 are from a cloud-based investment market with unpredictable investment opportunities because the projects may become unavailable between the time of making a decision and the time of implementation. Therefore, management is interested in considering projects from this cloud market (i.e. projects 1 and 2) in addition to their regular investment opportunities (i.e. projects 3 to 10). This article assumes that implementation uncertainties affect the projects in the cloud market and that the associated implemented decisions may differ from the decisions described by the optimization model; these variables are called ‘uncertain’. Furthermore, this work assumes that the implemented decisions associated with regular projects are the decisions described by the optimization model; these variables are called ‘deterministic’.

Because of implementation uncertainty, a prescribed solution’s profit and feasibility can be affected. Let \( x_i \) be the binary decision variable associated with the selection of project \( i \). Then, Table 2 shows all possible outcomes when implementing the deterministic optimal solution as prescribed; i.e. variables 3–10 values would remain as specified, while variables 1 and 2 may take any value during implementation. In this article, a solution to a BLP under implementation uncertainty is a set of solutions specified by the values of the deterministic variables. This work answers the question, ‘Is there a robust solution set with the desired objective values to guarantee the desired feasibility level?’ As will be described in Section 4.2 of the article, once this set is determined, an optimization problem may be formulated over the robust solution set to single out a solution for implementation.

Table 2 shows that the worst-case profit occurs when projects 1 and 2 are cancelled, leaving only the regular projects that are not affected by implementation uncertainty, yielding a lower profit of 34 instead of the optimal 41. Regarding feasibility, two of the four possible outcomes exceed the

### Table 1. Profits and costs for the candidate projects in the illustrative example.

| \( i \) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| \( c_i \) | 7 | 3 | 9 | 9 | 10 | 7 | 4 | 2 | 6 | 2 |
| \( a_i \) | 4 | 5 | 9 | 8 | 4 | 4 | 6 | 6 | 2 | 3 |

### Table 2. All possible outcomes after implementing the optimal deterministic solution. The prescribed solution is in bold font; italic font indicates infeasibility of the deterministic version of the problem.

| \( i \) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Profit | Required budget |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \( x_i \) | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 34 | 22 |
| 0 | 1 | 37 | 27 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 41 | 26 |
| 1 | 1 | 44 | 31 |
Table 3. Robust solution obtained with the proposed method. The prescribed solution is shown in bold font; italic font indicates infeasibility of the deterministic version of the problem.

| i   | x_i | Profit | Required budget |
|-----|-----|--------|-----------------|
| 1   | 0   | 32     | 18              |
| 2   | 0   | 35     | 23              |
| 3   | 1   | 0      | 1               |
| 4   | 1   | 0      | 1               |
| 5   | 1   | 0      | 1               |
| 6   | 0   | 39     | 22              |
| 7   | 0   | 42     | 27              |

budget owing to uncertainty, with one even producing a smaller profit. In general, implementation uncertainty may cause infeasibilities during implementation; hence, solution methodologies to tackle optimization problems under this type of uncertainty must explicitly consider possible feasibility violations (with respect to the deterministic problem)—in this article, adding a parameter $\delta_j \geq 0$ to the right-hand side of every constraint controls the level of acceptable infeasibility.

The problem studied in this article aims at finding solutions with close-to-optimal profit that guarantee a controlled infeasibility level. Table 3 displays the solution set found using the methodology proposed in Section 4.1 and allowing a maximum of 5% violation of the budget constraint; the method proposed in Section 4.2 specifies one solution ($\hat{x}_D$) from the robust set. Notice the differences between the deterministic and robust solutions in the variables associated with projects that will not change during implementation, i.e. projects 3 through 10. The robust solution displays lower profit than the deterministic optimal value (39 versus 41) but better feasibility performance (i.e. 3 out of 4 versus 2 out of 4 feasible outcomes). The proposed methodology was able to find a feasible solution where project 1 switches from 1 to 0 and project 2 is selected (i.e. switching from 0 to 1)—this situation was infeasible when using the optimal deterministic solution (see Table 2).

The robust solution’s feasibility seems more ‘controlled’, as evidenced by a maximum exceedance of the budget of one unit versus five units for the deterministic solution. Arguably, often in practice, a ‘small-controlled’ feasibility violation would be ‘easier’ to fix during implementation. In this example, if the infeasibility were to occur during implementation, it would be easier for the company to request a small increase in the budget to accommodate a good investment opportunity, e.g. to request an increase in one unit of budget, achieving a profit of 42; it is reasonable to assume that if the budget violation were larger, it would be more difficult to resolve in practice.

The proposed approach can also handle situations when deterministic constraint violations are not allowed. However, these worst-case solutions may significantly degrade the objective value compared to the deterministic optimal. Practical problems requiring these extreme worst-case solutions include decisions related to safety, loss of life, or negative impact on large populations of people, e.g. electricity network reliability, the design of vaccine distribution systems, complex surgery procedure planning, the design of anti-terrorism systems, warfare planning, etc.

1.3. Literature review

Different approaches aim to protect the optimality and feasibility of solutions in the face of uncertainties including stochastic optimization—e.g. Dantzig (1955), Beale (1955) and Wets (1966, 1974, 1983)—and robust optimization—e.g. Soyster (1973), Mulvey, Vanderbei, and Zenios (1995) and Bertsimas and Sim (2004). Stochastic optimization seeks solutions that remain optimal and feasible with high probability. However, there may exist realizations of the uncertainty where the optimality or feasibility are not satisfied—see Ben-Tal, Ghaoui, and Nemirovski (2009). On the other hand, robust optimization approaches seek solutions that satisfy the given levels of optimality and feasibility for any realization of the uncertainty; such solutions are termed robust solutions (Mulvey, Vanderbei, etc.).
and Zenios 1995). For instance, Soyster (1973) considers perturbations in the coefficients of the constraints using convex sets; the resulting model produces solutions that are feasible for any realization of the data within the convex sets.

The existing work in the field of robust optimization accounting for implementation uncertainty is very limited (Gabrel, Murat, and Thiele 2014). Ben-Tal, Ghaoui, and Nemirovski (2009) propose two forms of modelling implementation uncertainty on real decision variables: additive implementation errors refer to the case when a random value is added to the prescribed value, and multiplicative implementation error refers to the case when the random value multiplies the prescribed value; furthermore, the authors show that these forms of implementation errors are equivalent to artificial data uncertainties and can be treated as such. These forms of modelling implementation uncertainty have been used in single optimization problems—e.g. Das (1997) and Lewis and Pang (2009)—and in multiobjective optimization problems—e.g. Deb and Gupta (2006), Jornada and Leon (2016) and Eichfelder, Krüger, and Schöbel (2017). However, these models of implementation uncertainty cannot be extended to the case of binary problems when dealing with implementation uncertainty because using equivalent data uncertainty is not straightforward; additional discussion is provided in Section 2.

The methodology in this article assumes worst-case objective functions—see Kouvelis and Yu (1997) and Kang, Lee, and Lee (2012); the study of models with other types of objective function remains for future research. A characteristic of worst-case robust solutions is that they tend to be conservative because they may excessively sacrifice optimality to satisfy the given feasibility level. For instance, the model in Soyster (1973) is considered too conservative from this perspective (Bertsimas and Sim 2004). Different authors have addressed this issue by modelling uncertainty using different representations; for instance El Ghaoui and Lebret (1997), El Ghaoui, Oustry, and Lebret (1998) and Ben-Tal and Nemirovski (1998, 1999, 2002) propose a less conservative model by using ellipsoidal sets to describe data uncertainty, Bertsimas and Sim (2003, 2004) control conservatism by bounding the maximum number of uncertain coefficients changing in each constraint simultaneously, and Kouvelis and Yu (1997) use measures of robustness that seek to minimize the difference between the objective and robust objective values (i.e. the maximum deviation).

To address the conservatism of the robust solutions, the proposed robust formulation includes a feasibility parameter that gives the decision maker direct control of the feasibility level to improve the robust solutions’ objective performance. In addition, the proposed methodology also incorporates cardinality-constrained concepts inspired by the work of Bertsimas and Sim (2004) applied to implementation uncertainty directly affecting decision variables rather than model parameters.

The rest of the article is organized as follows. Section 2 describes the model for robust optimization under implementation uncertainty; Section 3 presents the formulation of the problem and the characteristics of the robust solution set; Section 4 describes the solution methodology to find robust solution sets and how to select solutions for implementation; Section 5 presents a combination of the proposed robust solution model with a cardinality-constrained approach; Section 6 presents an experimental study in the context of the knapsack problem, and Section 7 contains concluding remarks.

2. Model development

Consider the following BLP:

\[
\min \quad f(x) = \sum_{i=1}^{n} c_i x_i \\
\text{subject to} \quad \sum_{i=1}^{n} a_{ij} x_i \leq b_j, \quad \text{for } j = 1, \ldots, m.
\]
\( x_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, n. \) 

Henceforth, this formulation is called the \textit{deterministic BLP formulation}. Let \( \mathbb{B}^n = \{ x_i \in \mathbb{R} : x_i \in [0, 1) \text{ for } i = 1, \ldots, n \} \) be the set of binary solutions in \( \mathbb{R}^n \). The set of solutions satisfying (2) and (3), \( X = \{ x \in \mathbb{B}^n : \sum_{i=1}^n a_{ij} x_i \leq b_j, \text{ for } j = 1, \ldots, m \} \) is termed the \textit{deterministic feasible set}.

Let \( \hat{x} \) denote a prescribed solution and \( \tilde{x} \) denote an implemented solution. It is assumed that \( \hat{x} \) is the solution obtained from solving an optimization model, and \( \tilde{x} \) is the solution implemented in reality, which may be different from \( \hat{x} \) owing to implementation uncertainty. For example, consider a formulation where \( x_i \) is one if project \( i \) is selected and zero otherwise; further, assume that the optimization model prescribes the decision maker to choose project \( i \) (i.e. \( \hat{x}_i = 1 \)). If implementation uncertainty is considered to affect \( x_i \), it is possible that the implemented value of \( \tilde{x}_i \) unexpectedly changes to zero. Definition 2.1 formalizes this type of uncertainty.

\textbf{Definition 2.1:} Let \( p_i \) and \( q_i \) be the probabilities that prescribed and implemented values of \( x_i \) are equal if the prescribed value is zero or one, respectively. A binary variable \( x_i \) is under implementation uncertainty if either \( p_i < 1 \) or \( q_i < 1 \), and the following conditional probabilities hold:

\[
\begin{align*}
P(\tilde{x}_i = \hat{x}_i | \hat{x}_i = 0) &= p_i, \quad P(\tilde{x}_i = 1 - \hat{x}_i | \hat{x}_i = 0) = 1 - p_i, \\
P(\tilde{x}_i = \hat{x}_i | \hat{x}_i = 1) &= q_i, \quad P(\tilde{x}_i = 1 - \hat{x}_i | \hat{x}_i = 1) = 1 - q_i.
\end{align*}
\] 

Ben-Tal, Ghaoui, and Nemirovski (2009) propose the additive and multiplicative implementation errors to model implementation uncertainty in real variables. The additive implementation errors consist of a random value \( \epsilon \) added to the prescribed value of the decision variable, \( \tilde{x} = \hat{x} + \epsilon \). On the other hand, the multiplicative implementation errors consist of a random value multiplying the prescribed value of the decision variable, \( \tilde{x} = \epsilon \hat{x} \). In these two models, the value of \( \epsilon \) belongs to a defined set. However, these models of implementation uncertainty cannot handle the case of binary variables appropriately. For instance, in the case of additive implementation errors, and assuming \( \epsilon \in \{-1, 0, 1\} \), adding \( \epsilon \) to the value decision variable may generate infeasible values for the decision variable. For example, with \( \hat{x} = 0 \), the implemented value \( \tilde{x} \) could be \(-1, 0 \text{ or } 1\); similarly, with \( \hat{x} = 1 \) the implemented value \( \tilde{x} \) may be \( 0, 1 \text{ or } 2\). Similarly, in the case of multiplicative implementation error with \( \hat{x} = 0 \), the result is \( \tilde{x} = \epsilon \hat{x} = 0 \) for any value of \( \epsilon \); hence, neither of these models of implementation uncertainty is appropriate.

Section 3 presents a model to handle the impact of implementation uncertainty in binary variables. Section 4 presents a solution method for the proposed model. The rest of this section presents several concepts, assumptions and notation used throughout the article.

Variables affected by implementation uncertainty are termed \textit{uncertain variables}, otherwise they are termed \textit{deterministic variables}. Without loss of generality, the decision vector \( \hat{x} \) is decomposed into two vectors \( x_C \) and \( x_U \), where \( x_C \) is composed of the deterministic variables \( x_1, \ldots, x_C \), and \( x_U \) is composed of the uncertain variables \( x_{C+1}, \ldots, x_n \); for convenience, define \( C = \{1, \ldots, C\} \) as the set of indices of the deterministic variables in \( x_C \), and \( U = \{C + 1, \ldots, n\} \) as the set of indices of the uncertain variables in \( x_U \). In the example in Section 1.2, projects 1 and 2 are associated with uncertain variables, while the remaining projects are related to deterministic variables.

\textbf{Assumption 2.1:} Sets \( C \) and \( U \) are given; \textit{i.e.} it is known which variables are deterministic and uncertain.

\textbf{Assumption 2.2:} Probabilities \( p_i \) and \( q_i \) for the uncertain variables are unknown; \textit{i.e.} it is known that the values of the uncertain variables may change during implementation, but the probability that a change may occur is unknown.

For a solution \( \hat{x} = (\hat{x}_C, \hat{x}_U) \), a corresponding implemented solution \( \tilde{x} = (\tilde{x}_C, \tilde{x}_U) \) has the same values of the deterministic variables; \textit{i.e.} \( \tilde{x}_C = \hat{x}_C \). On the other hand, the value \( \tilde{x}_U \) takes any possible
The set containing all the possible implemented solutions is called the set of implemented outcomes and is denoted by \( \mathcal{U}(x) \).

**Definition 2.2:** The set of possible implemented outcomes for a given solution \( x = (x_C, x_U) \in \mathbb{B}^n \) is defined as \( \mathcal{U}(x) = \{ \tilde{x} = (\tilde{x}_C, \tilde{x}_U) \in \mathbb{B}^n : \tilde{x}_C = x_C, \tilde{x}_U \in \{0, 1\} \} \).

The number of possible implementation outcomes grow exponentially with the number of uncertain variables; i.e. \( |\mathcal{U}(x)| = 2^{l(x)} \).

### 3. RBIU-\( \delta \) Problem Formulation

When impacted by implementation uncertainty, the objective value at implementation may differ from the prescribed objective, or the implemented solution may become infeasible. Therefore, a robust BLP under implementation uncertainty (RBIU-\( \delta \)) aims at finding solutions that guarantee desired levels of optimality and feasibility in the face of implementation uncertainty.

The **objective robustness level**, \( \gamma(x) \), measures the degree to which a solution’s objective function value degrades when affected by implementation uncertainty. \( \gamma(x) \) provides the worst-case (maximum) value of the objective function among all outcomes in \( \mathcal{U}(x) \), guaranteeing that the objective value of the implemented solution does not worsen when affected by implementation uncertainty; i.e. \( \gamma(x) \) is an upper bound for the objective function value when affected by uncertainty.

**Definition 3.1:** Given a binary vector \( x \), the objective robustness level, \( \gamma(x) \), is defined as

\[
\gamma(x) = \max_{y \in \mathcal{U}(x)} \{ f(y) \}. \tag{5}
\]

An advantage of the worst-case objective produced by \( \gamma(x) \) is that it allows linearizing expression (5) (see Section 4).

In robust optimization, it is common to obtain very conservative solutions (i.e. the resulting degradation in objective function may be too excessive) when desiring to preserve feasibility for the deterministic version of the problem (this work refers to this feasibility as **deterministic feasibility**). A better objective performance can be obtained in robust optimization by accepting some degree of infeasible outcomes. For instance, the cardinality-constrained approach shown in Bertsimas and Sim (2004) achieves less conservative solutions by limiting the number \( \Gamma \) of coefficients that may simultaneously be affected by uncertainty. However, this formulation may produce infeasible solutions when more than \( \Gamma \) coefficients are affected by uncertainty. A successful application of this approach relies on proving a low probability of the occurrence of infeasibilities. Unfortunately, this approach does not explicitly offer any guarantee in the level of deterministic feasibility violation, which may be problematic when dealing with binary variables affected by implementation uncertainty. Similar to the ‘feasibility tolerance’ concept in Ben-Tal and Nemirovski (2000), this work deals with the feasibility violation problem by introducing the feasibility parameter, \( \delta_j \geq 0 \), to guarantee a maximum level of constraint violation while expanding the robust solution space to include solutions with better objective performance.

The deterministic constraints (2) of the BLP are reformulated to protect against the maximum value produced by uncertainty, i.e. to guarantee that the maximum value produced by \( \sum_{i=1}^{n} a_{ij}x_i \) does not exceed \( b \) when impacted by implementation uncertainty. Incorporating the parameter \( \delta_j \), the constraints (2) are reformulated as follows:

\[
\max_{y \in \mathcal{U}(x)} \left\{ \sum_{i=1}^{n} a_{ij}y_i \right\} \leq b_j + \delta_j, \quad \text{for } j = 1, \ldots, m. \tag{6}
\]

The resulting formulation, called RBIU-\( \delta \), seeks to minimize the worst-case objective value of \( \gamma(x) \), i.e. to improve the objective robustness level among the feasible set defined by the modified
Figure 1. Illustration of the feasible sets $X$ and $\mathcal{X}$: (a) illustrates the most conservative case when $\delta_j = 0$ for all $j \in J$; (b) illustrates a less conservative case when there exists $j$ such that $\delta_j > 0$.

constraints (6). RBIU-$\delta$ is formulated as follows:

$$\min \gamma(x) = \max_{y \in \mathcal{U}(x)} \{f(y)\}$$

subject to $\max_{y \in \mathcal{U}(x)} \left\{ \sum_{i=1}^{n} a_{ij}y_i \right\} \leq b_j + \delta_j$, for $j = 1, \ldots, m.$

(8)

$$x_i \in \{0,1\}, \text{ for } i = 1, \ldots, n.$$  

(9)

Let $\mathcal{X}$ denote the feasible region defined by (8) and (9). By setting $\delta_j = 0$, for all $j \in J$, the RBIU-$\delta$ guarantees solutions, if they exist, that are feasible with respect to the deterministic problem, i.e. $\mathcal{X} \subseteq X$ (Figure 1(a)). However, the deterministic solution may not belong to $\mathcal{X}$, leading to a degradation of the objective value. This case is akin to other conservative robust approaches—e.g. that of Soyster (1973). On the other hand, by setting at least one $\delta_j > 0$, the RBIU-$\delta$ expands the robust feasible region $\mathcal{X}$ to include solutions that may or may not be feasible for the deterministic problem (Figure 1(b)). The expansion of $\mathcal{X}$ seeks to evaluate solutions that may have a more attractive objective value when affected by implementation uncertainty, as opposed to the cardinality-constrained formulations restricted to solutions in $X$ only. Section 5 presents a formulation that combines the RBIU and cardinality-constrained formulations, RBIU-$\delta$-CC, taking advantage of both methods and allowing the decision maker to control the degree of the conservatism of the solutions while guaranteeing a desired level of deterministic constraint violation.

RBIU-$\delta$ possesses multiple optimal solutions given by $\mathcal{U}(x^*)$ (see Property 3.2). This set is called the robust optimal solution set and denoted by $\mathcal{U}^*$. It is only necessary to determine $x_c$ to find the set $\mathcal{U}^*$. Although the solutions in $\mathcal{U}^*$ may produce different values of the objective function, none of them exceeds $\gamma(x)$.

**Property 3.2:** Let $x^* = (x^*_C, x^*_U)$ be an optimal solution to the RBIU-$\delta$. Any solution $y = (y_C, y_U) \in \mathcal{U}(x^*)$ is also optimal to the RBIU-$\delta$.

Ramirez Calderon (2018) formalizes Property 3.2. The following section presents the solution methodology for the RBIU-$\delta$.

4. Solution methodology

The proposed methodology to solve the RBIU-$\delta$ consists of two stages. Stage I finds, if it exists, a robust optimal solution set $\mathcal{U}^*$. Section 4.1 presents a method for finding $\mathcal{U}^*$. However, from a practical perspective, finding a specific solution from $\mathcal{U}^*$ is of interest. Therefore, stage II consists of selecting a solution from $\mathcal{U}^*$ with desired properties; Section 4.2 shows the method for choosing such a solution.
4.1. Binary linear programming reformulation of the RBIU-\(\delta\)

This work proposes reformulating the RBIU-\(\delta\) into a binary linear programming problem that requires solving a similar problem with fewer variables than the original deterministic BLP formulation.

Reformulating the RBIU-\(\delta\) into a binary linear programming problem requires finding linear expressions equivalent to ones with max operators in (7) and (8). Lemma 4.1 proposes the equivalent linear expressions for these maximum values.

**Lemma 4.1:** Given a binary vector \(x = (x_1, \ldots, x_n)\) formed by deterministic variables \(x_1, \ldots, x_c\) and uncertain variables \(x_{c+1}, \ldots, x_n\), the following equalities hold:

\[
\max_{y \in \mathcal{U}(x)} \{f(y)\} = \sum_{i \in C} c_i x_i + \sum_{i \in U} (c_i + |c_i|)/2 \tag{10}
\]

\[
\max_{y \in \mathcal{U}(x)} \left\{ \sum_{i=1}^{n} a_{ij} y_i \right\} = \sum_{i \in C} a_{ij} x_i + \sum_{i \in U} (a_{ij} + |a_{ij}|)/2, \quad \text{for } j = 1, \ldots, m. \tag{11}
\]

**Proof:** For equality (10), it suffices to prove that \((c_i + |c_i|)/2 > c_i x_i\) for all \(i\), i.e. \((c_i + |c_i|)/2\) is the maximum value of \(c_i x_i\) for any value of \(x_i\). If \(c_i \geq 0\), then \(|c_i| = c_i\) and \((c_i + |c_i|)/2 = (c_i + c_i)/2 = c_i \geq c_i x_i\); similarly, if \(c_i < 0\), then \(|c_i| = -c_i\) and \((c_i + |c_i|)/2 = (c_i - c_i)/2 = 0 \geq c_i x_i\). Therefore, \((c_i + |c_i|)/2\) is the maximum value of \(c_i x_i\) for all \(i\).

Because the values of the deterministic variables \(x_1, \ldots, x_c\) do not change, then \(\max_{y \in \mathcal{U}(x)} \{f(y)\} = \sum_{i \in C} c_i x_i + \max_{y \in \mathcal{U}(x)} \{\sum_{i \in U} c_i y_i\} = \sum_{i \in C} c_i x_i + \sum_{i \in U} (c_i + |c_i|)/2\).

Equality (11) can be proved following the same rationale.

The binary linear programming reformulation of the RBIU-\(\delta\) is the following:

\[
\min \gamma(x) = \sum_{i \in C} c_i x_i + \sum_{i \in U} (c_i + |c_i|)/2 \tag{12}
\]

subject to

\[
\sum_{i \in C} a_{ij} x_i + \sum_{i \in U} (a_{ij} + |a_{ij}|)/2 \leq b_j + \delta_j, \quad \text{for } j = 1, \ldots, m. \tag{13}
\]

\[
x_i \in \{0, 1\}, \quad \forall i \in C. \tag{14}
\]

This binary linear programming reformulation finds, if it exists, one solution of the RBIU-\(\delta\), called \(x^* = (x^*_C, x^*_U)\), and therefore also the robust optimal solution set \(\mathcal{U}(x^*)\). In addition, this reformulation depends on \(|C| < n\) deterministic variables only because constant values replace the uncertain variables, thus reducing the search space.

Figure 2 shows a flow diagram with the procedure to formulate the RBIU-\(\delta\) associated with a BLP under implementation uncertainty.

The resulting reformulation of RBIU-\(\delta\) is a BLP that can be solved using existing methodologies and tools; for instance, CPLEX® and AMPL®.

4.2. Selecting specific robust optimal solutions

Stage I of the solution of the RBIU-\(\delta\) produces a robust optimal solution set containing multiple solutions. Stage II allows the decision maker to select one solution from the set with specific characteristics.

Given a set \(\mathcal{U}(x^*)\) containing the robust solutions \(x^* = (x^*_C, x^*_U)\), stage II consists of solving the deterministic BLP formulation (1), (2) and (3) and setting its deterministic variables to \(x^*_C\) to ensure membership in the robust optimal solution set. As a result, the selected solution \(\hat{x}\) will have the
desired robustness regarding levels of objective value, feasibility performance and other characteristics that make it desirable for implementation. Table 4 presents several different specific robust solutions selected from \( U(x^*) \).

Selection problem SP1 attempts to find the solution \( \hat{x}^D \) that satisfies constraints (2) when the values of the deterministic variables are set equal to those of the deterministic variables associated with \( U(x^*) \). This solution, if it exists, has the best objective function and is feasible for the deterministic BLP formulation since it considers feasible solutions in the deterministic feasible region \( X \).

Selection problem SP2 attempts to find the solution \( \hat{x}^R \) whose deterministic variables are set equal to those of the deterministic variables associated with \( U(x^*) \). The feasible region is relaxed to allow some infeasibilities for the deterministic BLP formulation. This solution, if it exists, may have a better objective function than a solution produced by SP1. Still, it may be infeasible for the deterministic BLP formulation since it considers solutions in the robust feasible region \( X^* \) instead of \( X \).

Selection problems SP3 and SP4 attempt to select robust solutions \( \hat{x}^{UB} \) and \( \hat{x}^{LB} \) yielding the highest and lowest objective function value for any solution in \( U(x^*) \), respectively. The following proposition allows identifying these solutions in a given \( U(x^*) \) in linear time.

**Proposition 4.2:** Given a robust optimal solution set \( U(x^*) \), a robust solution \( \hat{x}^{UB} \in U(x^*) \) produces an upper bound for the objective function value by setting its values as follows:

\[
\hat{x}^{UB}_i = \begin{cases} 
  x^*_i & \text{if } i \in C \\
  1, & \text{if } c_i \geq 0, \ i \in U \\
  0, & \text{if } c_i < 0, \ i \in U.
\end{cases}
\] (15)
Similarly, a robust solution $\hat{x}^{LB} \in U(x^*)$ produces a lower bound for the objective function value by setting its values as follows:

$$
\hat{x}_i^{LB} = \begin{cases} 
  x_i^* & \text{if } i \in C \\
  0 & \text{if } c_i \geq 0, \ i \in U \\
  1 & \text{if } c_i < 0, \ i \in U.
\end{cases}
$$

(Ramirez Calderon (2018) presents the proof of Proposition 4.2.)

The performance of these selected solutions will be illustrated later in a case study in the context of the knapsack problem.

5. Combined feasibility parameter and cardinality-constrained robust formulation, RBIU-$\delta$-CC

As mentioned above, this work presents a formulation that combines the RBIU-$\delta$ and cardinality-constrained formulations—see Bertsimas and Sim (2004)—denoted by RBIU-$\delta$-CC. This formulation aims to allow the decision maker to control the degree of the conservatism of the solutions in addition to the desired level of deterministic constraint violation. In contrast to Bertsimas and Sim’s robust formulation, RBIU-$\delta$-CC considers uncertainty affecting the variables instead of the coefficients of the model and impacting the entire column of the uncertain variables, including constraints and the objective function simultaneously. Moreover, the interval for uncertainty in the proposed model is asymmetric and binary owing to the nature of the decision variables, in contrast to a symmetric and real interval in Bertsimas and Sim (2004).

Consider expressions (10) and (11) in Lemma 4.1. Let $\Gamma$ be an integer control parameter, called the cardinality-constrained parameter, such that $1 \leq \Gamma \leq |U|$. $\Gamma$ represents the maximum number of uncertain variables that simultaneously may have different prescribed and implemented values. The RBIU-$\delta$-CC is formulated as follows:

$$
\text{min } \gamma
$$

subject to

$$
\sum_{i \in C} c_i x_i + \max_{\{S_0: S_0 \subseteq U, |S_0| \leq \Gamma\}} \left\{ \sum_{i \in S_0} \frac{c_i + |c_i|}{2} + \sum_{i \in U \setminus S_0} c_i x_i \right\} \leq \gamma
$$

$$
\sum_{i \in C} a_{ij} x_i + \max_{\{S_j: S_j \subseteq U, |S_j| \leq \Gamma\}} \left\{ \sum_{i \in S_j} \frac{a_{ij} + |a_{ij}|}{2} + \sum_{i \in U \setminus S_j} a_{ij} x_i \right\} \leq b_j + \delta_j,
$$

for $j = 1, \ldots, m$

$$
x_i \in \{0, 1\}, \ i = 1, \ldots, n.
$$

(RBIU-$\delta$-CC protects the optimality and feasibility levels against any, at most $\Gamma$, uncertain variables with different prescribed and implemented values.

The selection of the at most $\Gamma$ uncertain variables requires the enumeration of all the subsets $S_0$ and $S_j$ of $U$ whose cardinality is less than or equal to $\Gamma$. The development of an equivalent linear reformulation to the RBIU-$\delta$-CC follows the work in Bertsimas and Sim (2004), and it is presented in detail in Ramirez Calderon (2018). The RBIU-$\delta$-CC is equivalent to the following mixed-integer linear programming (MIP) reformulation:

$$
\text{min } \gamma
$$

subject to

$$
\sum_{i \in C} c_i x_i + \Gamma v_0 + u_{00} + \sum_{i \in U} u_{i0} \leq \gamma
$$
\begin{align*}
  v_0 + u_{i0} &\geq \frac{c_i + |c_i|}{2} - c_jx_i, \quad \forall i \in U \\
  u_{00} &\geq \sum_{i \in U} c_ix_i \\
  \sum_{i \in C} a_{ij}x_j + \Gamma v_j + u_{0j} + \sum_{i \in U} u_{ij} &\leq b_j + \delta_j, \quad j = 1, \ldots, m \\
  v_j + u_{ij} &\geq \frac{a_{ij} + |a_{ij}|}{2} - a_{ij}x_i, \quad \forall i \in U, \ j = 1, \ldots, m \\
  u_{0j} &\geq \sum_{i \in U} a_{ij}x_i, \quad j = 1, \ldots, m \\
  u_{i0}, u_{ij}, v_0, v_j &\geq 0, \quad \forall i \in U, \ j = 1, \ldots, m \\
  u_{00}, u_{0j} &\text{ are unrestricted} \quad j = 1, \ldots, m \\
  x_j &\in \{0, 1\}, \quad i = 1, \ldots, n.
\end{align*}

RBIU-\(\delta\)-CC produces more optimistic solutions than RBIU-\(\delta\) at the expense of possible violation of the desired feasibility level or degradation of the optimality when more than \(\Gamma\) uncertain variables are affected simultaneously.

Similarly to RBIU-\(\delta\), RBIU-\(\delta\)-CC produces a robust optimal solution set \(\mathcal{U}(x^*)\). Hence, the method shown in Section 4.2 for selecting specific robust optimal solutions applies to the set result of RBIU-\(\delta\)-CC. In other words, RBIU-\(\delta\)-CC is another method for completing stage I of the solution of a robust formulation. The RBIU-\(\delta\)-CC becomes RBIU-\(\delta\) when \(\Gamma = |U|\).

Figure 3 shows a flow diagram with the procedure for formulating the RBIU-\(\delta\) associated with a BLP under implementation uncertainty.

As mentioned before, RBIU-\(\delta\)-CC may become infeasible if more than \(\Gamma\) uncertain variables change their values during implementation. Ramirez Calderon (2018) presents theoretical upper bounds for the probability that the RBIU-\(\delta\)-CC becomes infeasible due to violation of the \(\Gamma\) limit.

6. Case study: the knapsack problem under implementation uncertainty

This section presents a case study of applications of the formulations and methodologies presented in previous sections in the context of the knapsack problem (KP). The objective of the case study is (1) to exemplify the application of the proposed robust formulations, and (2) to exemplify the characteristics of the solutions that are robust to implementation uncertainty.

The KP is a well-known binary linear programming problem with applications in cryptography, manufacturing, economy, etc. The example in Section 1.2 shows a particular application of this problem. The similarity of its structure to the BLP described in Section 2 makes it an adequate application to exemplify the proposed work.

6.1. Implementation-robust knapsack problem

6.1.1. Formulation

This section shows how to model the KP under implementation uncertainty using the proposed robust formulations. Given a set of \(n\) options, each with return \(c_i\) and weight \(a_i\), the objective of the KP is to determine which options should be selected to maximize profit while the total weight of the chosen options does not exceed a budget \(b\). The KP is formulated as the following BLP:

\[
\max \sum_{i=1}^{n} c_ix_i
\]
Subject to
\[ \sum_{i=1}^{n} a_i x_i \leq b \] (32)

\[ x_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, n. \] (33)

This formulation will be referred to as the deterministic KP. The objective \( \min \sum_{i=1}^{n} -c_i x_i \) replaces (31) to coincide with the minimization objective used in previous sections noting that it provides a negative value of (31).

Assuming that the deterministic KP possesses at least one uncertain variable (as in the previous sections, deterministic and uncertain variables are defined by sets \( C \) and \( U \), respectively), the associated robust formulation of the KP under implementation uncertainty (RKP-\( \delta \)) according to the RBU-\( \delta \) in (7), (8) and (9) is the following:

\[
\min \gamma(x) = \max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i=1}^{n} -c_i y_i \right\} \tag{34}
\]

subject to
\[
\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i=1}^{n} a_i y_i \right\} \leq b + \delta \tag{35}
\]

\[ x_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, n. \] (36)

Given the objective \( \min \sum_{i=1}^{n} -c_i x_i \), the diagram in Figure 2 can be used to compute the BLP reformulation as follows.

**Step 1:** Compute

\[
(c_i + |c_i|)/2 = (-c_i + |c_i|)/2 = (-c_i + c_i)/2 = 0, \quad \forall i \in U. \tag{37}
\]
Step 2: Compute

\[
\frac{(a_i + |a_i|)}{2} = \frac{(a_i + a_i)}{2} = a_i, \quad \forall i \in U.
\] (38)

Step 3: Replace \(c_i x_i\) by (37) for all \(i \in U\).
Step 4: Replace \(a_i x_i\) by (38) for all \(i \in U\).
Step 5: Replace \(b\) by \(b + \delta\).

The reformulation of the RKP-\(\delta\) as a BLP is as follows:

\[
\min \gamma(x) = \sum_{i \in C} -c_i x_i
\] (39)

subject to

\[
\sum_{i \in C} a_i x_i + \sum_{i \in U} a_i \leq b + \delta,
\] (40)

\[
x_i \in \{0, 1\}, \quad \forall i \in C.
\] (41)

By transforming this reformulation into a maximization problem, the RKP-\(\delta\) can be rewritten as follows:

\[
\max \sum_{i \in C} c_i x_i
\] (42)

subject to

\[
\sum_{i \in C} a_i x_i \leq b'
\] (43)

\[
x_i \in \{0, 1\}, \quad \forall i \in C.
\] (44)

where \(b' = b + \delta - \sum_{i \in U} a_i\). This formulation corresponds to a knapsack problem formulation with fewer variables than the deterministic KP. Moreover, it depends on deterministic variables only, making it a deterministic KP itself. This result shows a virtue of the proposed robust formulation and its linearization; as mentioned before, it reduces the optimization problem at hand.

Similarly, the RBIU-\(\delta\)-CC can be used to deal with implementation uncertainty in the KP. This formulation, denoted by RKP-\(\delta\)-CC, is as follows:

\[
\min \gamma
\] (45)

s.t.

\[
\sum_{i \in C} -c_i x_i + \max_{S : |S| \leq \Gamma} \left\{ \sum_{i \in S} \left( -\frac{c_i + |c_i|}{2} \right) + \sum_{i \in U \setminus S} c_i x_i \right\} \leq \gamma(x)
\] (46)

\[
\sum_{i \in C} a_i x_i + \max_{S : |S| \leq \Gamma} \left\{ \sum_{i \in S} \left( -\frac{a_i + |a_i|}{2} \right) + \sum_{i \in U \setminus S} a_i x_i \right\} \leq b + \delta, \quad \forall j \in J
\] (47)

\[
x_i \in \{0, 1\}, \quad \forall i \in I.
\] (48)

The RKP-\(\delta\)-CC can be solved using the equivalent linear reformulation in Section 5. For brevity, this work omits the reformulation of the RKP-\(\delta\)-CC.

6.1.2. Illustrative example

Consider the illustrative example in Section 1.2. As mentioned before, this problem corresponds to a knapsack problem. The corresponding deterministic KP formulation given the data in Table 1 is as follows:

\[
\max 7x_1 + 3x_2 + 9x_3 + 9x_4 + 10x_5 + 7x_6 + 4x_7 + 2x_8 + 6x_9 + 2x_{10}
\] (49)
subject to  \[ 4x_1 + 5x_2 + 9x_3 + 8x_4 + 4x_5 + 4x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10} \leq 26 \]  
\[ x_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, 10. \]  

The objective of this formulation transformed as a minimization problem is as follows:

\[ \min -7x_1 - 3x_2 - 9x_3 - 9x_4 - 10x_5 - 7x_6 - 4x_7 - 2x_8 - 6x_9 - 2x_{10}. \]

The corresponding RKP-\( \delta \) as shown in (34), (35) and (36) is as follows:

\[
\begin{align*}
\min \quad & y(x) = \max_{y \in U(x_C)} \left\{ -7y_1 - 3y_2 - 9y_3 - 9y_4 - 10y_5 - 7y_6 - 4y_7 - 2y_8 - 6y_9 - 2y_{10} \right\} \\
\text{s.t.} \quad & \max_{y \in U(x_C)} \left\{ 4y_1 + 5y_2 + 9y_3 + 8y_4 + 4y_5 + 4y_6 + 6y_7 + 6y_8 + 2y_9 + 3y_{10} \right\} \leq 26 + \delta \\
& x_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, 10.
\end{align*}
\]

The illustrative example considers that implementation uncertainty may impact variables 1 and 2, making them uncertain. Following the diagram in Figure 2 the reformulation into a BLP is as follows.

**Step 1:** Computations (37) for variables 1 and 2 are the following:

\[
\begin{align*}
(c_1 + |c_1|)/2 &= (-7 + | - 7|)/2 = (-7 + 7)/2 = 0 \\
(c_2 + |c_2|)/2 &= (-3 + | - 3|)/2 = (-3 + 3)/2 = 0.
\end{align*}
\]

**Step 2:** Computations (38) for variables 1 and 2 are the following:

\[
\begin{align*}
(a_1 + |a_1|)/2 &= (4 + |4|)/2 = (4 + 4)/2 = 4 \\
(a_2 + |a_2|)/2 &= (5 + |5|)/2 = (5 + 5)/2 = 5.
\end{align*}
\]

**Step 3** Replace \( 7x_1 \) and \( 3x_2 \) by zero to obtain:

\[ 9x_3 + 9x_4 + 10x_5 + 7x_6 + 4x_7 + 2x_8 + 6x_9 + 2x_{10}. \]

**Step 4** Replace \( 4x_1 \) and \( 5x_2 \) by 4 and 5, respectively, to obtain:

\[ 9 + 9x_3 + 8x_4 + 4x_5 + 4x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10}. \]

**Step 5** Replace 26 by \( 26 + \delta \).

Finally, \( b' = b + \delta - \sum_{i\in U} a_i = 26 + \delta - 4 - 5 = 17 + \delta \). Therefore, the reformulation of the RKP-\( \delta \) to a deterministic RKP is as follows:

\[
\begin{align*}
\max \quad & 9x_3 + 9x_4 + 10x_5 + 7x_6 + 4x_7 + 2x_8 + 6x_9 + 2x_{10} \\
\text{s.t.} \quad & 9x_3 + 8x_4 + 4x_5 + 4x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10} \leq 17 + \delta \\
& x_i \in \{0, 1\}, \quad \text{for } i = 3, \ldots, 10.
\end{align*}
\]

### 6.2. Experimental study of robust solutions

This section shows a study of the solutions to the deterministic KP and its associated robust formulations when they are affected by implementation uncertainty. The objective is to exemplify the characteristics of the solutions and show their objective function and feasibility performances.
In particular, findings in this section show that a decision maker could use the RKP-\(\delta\) to protect the feasibility of the KP under implementation uncertainty, \(i.e.\) they could produce a solution that does not exceed the budget \(b\) even when impacted by uncertainty, with a low reduction of the profit (the objective value). Furthermore, the results reveal that, as more uncertainty impacts the problem, the decision maker could seek to increase the feasibility relaxation, \(i.e.\) increase the given budget by increasing the value of \(\delta\), to obtain a higher profit while guaranteeing that the budget will not be exceeded when impacted by uncertainty.

### 6.2.1. Design of the experiments

This section describes the experimental study used to exemplify the characteristics of robust solutions and how their performance is measured.

The problems used in this study consist of twelve instances of a knapsack problem with one hundred variables \((n = 100)\) each. The returns \(c_i\), weights \(a_i\) and budget \(b\) for each of the instances are generated as follows.

- Each \(c_i\) is randomly chosen from the set \([21, 22, \ldots, 80]\).
- Each \(a_i\) is randomly chosen from the set \([41, 42, \ldots, 60]\).
- Budget \(b\) is chosen as half of the total weight, \(i.e.\) \(b = 0.5 \sum_{i=1}^{n} a_i\).

The optimal solution and objective value \(f^*\) are determined for each instance using CPLEX. The optimal solution \(x^*\) is recorded to impact it by implementation uncertainty later.

The level of uncertainty, feasibility relaxation and cardinality-constrained parameters used in the experiment are defined as follows.

- The number of uncertain variables \(u\) defines the level of uncertainty. This study considers 1, 3, 5, 7 and 9 to be uncertain variables. Each uncertain variable may keep its value during implementation or change it.
- The feasibility relaxation \(\delta\) defines the maximum level of constraint violation. This study considers \(\delta\) equal to 0\%, 1\%, 2\%, 3\%, 4\% and 5\% of the value of the budget \(b\).
- The cardinality-constrained \(\Gamma\) defines the maximum number of uncertain variables that may be affected by implementation uncertainty simultaneously. This study considers \(\Gamma\) to be 40\%, 60\% and 80\% of the uncertain variables (rounding up). These values of \(\Gamma\) apply to all the values considered before except for \(u = 1\).

RKP-\(\delta\) and RKP-\(\delta\)-CC are solved using CPLEX. Given the solutions of the deterministic KP, RKP-\(\delta\) with the different values of \(\delta\), and RKP-\(\delta\)-CC with the different values of \(\Gamma\) and \(\delta\), each solution is impacted by implementation uncertainty. Because of implementation uncertainty, there exists a set of possible implemented solutions \(\mathcal{U}(x)\) containing all possible combinations of the different values that uncertain variables may take; in contrast, the deterministic variables keep the values defined by the solution. Therefore, this study utilizes a full enumeration of the elements of \(\mathcal{U}(x)\) to evaluate their performance using the following measures.

- The feasibility performance, \(h(x)\), is measured as the proportion of the solutions in set \(\mathcal{U}(x)\) that do not exceed the budget \(b\), \(i.e.\) they are feasible with respect to the deterministic KP.
- The objective value performance, \(\Delta(x)\), is measured as the proportion of the deterministic solution’s profit produced by the average profit of elements \(y \in \mathcal{U}(x)\).

A value of \(h(x)\) equal to one means that none of the possible outcomes resulting from implementation uncertainty exceeds the budget \(b\). Similarly, the greater the \(\Delta(x)\), the greater is the proportion of the deterministic solution’s profit produced by the elements of \(\mathcal{U}(x)\).
6.2.2. Analysis of the results for RKP-δ and RKP-δ-CC
This section shows the results and analysis of the performance of the deterministic and robust solutions when impacted by implementation uncertainty.

The results in this section reveal the following.

- RKP-δ produces solutions with more possible outcomes from implementation uncertainty not exceeding the budget \( b \) than the deterministic solution, i.e. RKP-δ possesses better feasibility performance than the deterministic KP.
- The feasibility performance of the RKP-δ is achieved with no significant reduction in the profit.
- The decision maker could seek to increase the budget by increasing \( \delta \) as uncertainty increases to obtain solutions with better profit while guaranteeing that the budget will not be exceeded.
- The RKP-δ-CC provides better profit while guaranteeing high feasibility performance for large values of \( \Gamma \).
- The interval for the RKP-δ objective value produced by the lower and upper bound solutions is small, showing that robust solutions produce objective values with low variability.

The results for the RKP-δ are shown in Table 5 and Figure 4. Table 5 shows the average of the objective value performance \( \Delta(x) \) for the twelve instances of the knapsack problem.
Similarly, Figure 4 shows the average of the feasibility performance $h(x)$ of each solution under implementation uncertainty for the twelve instances of the knapsack problem.

The results in Table 5 and Figure 4 indicate the following.

- Implementation uncertainty highly impacts the feasibility of the deterministic solution. There are 20% of the possible outcomes that exceed the given budget for one uncertain variable and this continues to increase as the number of uncertain variables increases, reaching 60% of the possible outcomes that exceed the budget.
- Robust solutions are attractive from a practical standpoint because the tradeoff between feasibility and objective function is insignificant. The RKP-$\delta$ for $\delta = 0\%$ guarantees solutions that do not exceed the budget for any result of uncertainty with little reduction of the profit. The decrease in the profit does not exceed 10%, even for a more significant number of uncertain variables.
- Figure 4 shows a trend in the behaviour of the RKP-$\delta$ as $\delta$ increase. The feasibility performance increases as the uncertainty increases; therefore, for greater levels of uncertainty, larger values of $\delta$ provide more protection against implementation uncertainty.
- Table 5 shows that greater $\delta$ values produce better profits, thus reducing the conservatism of the robust solutions. These results suggest that the decision maker could adjust the $\delta$ value according to the uncertainty level to obtain more attractive solutions.

Table 6 and Figure 5 show the results of the RKP-$\delta$-CC for $\delta = 0\%$ and different values of $\Gamma$. They also contain solutions of the deterministic KP and RKP-$\delta$ for $\delta = 0\%$ as a baseline. The results in Table 6 and Figure 5 indicate the following.

- As expected, small values of $\Gamma$ produce solutions with better profits; the objective value for $\Gamma = 40\%$ behaves almost like the deterministic solution. However, the tradeoff between feasibility and optimality is still present as the feasibility is highly impacted by more possible outcomes exceeding the budget.
- Table 6 also shows that, as the value of $\Gamma$ increases, the profit decreases, tending to behave like the RKP-$\delta$. Similarly, the feasibility increases as $\Gamma$ increases.
Table 6. Average of the objective value performance of the solutions for the deterministic KP, RKP-$\delta$ and RKP-$\delta$-CC under implementation uncertainty for $\delta = 0\%$ and different values of $\Gamma$ for the twelve instances of the knapsack problem.

| Formulation                        | Number of uncertain variables, $u$ |
|------------------------------------|------------------------------------|
|                                    | $u = 3$                           | $u = 5$                           | $u = 7$                           | $u = 9$                           |
| Deterministic                      | 0.994,518,15                      | 0.996,384,812                     | 0.987,165,005                     | 0.978,865,513                     |
| RKP-$\delta$ with $\delta = 0\%$ | 0.970,353,468                     | 0.948,841,825                     | 0.929,545,441                     | 0.909,664,756                     |
| RKP-$\delta$-CC with $\delta = 0\%$ and $\Gamma = 40\%$ | 0.625                             | 0.625                             | 0.563,802,083                     | 0.502,604,167                     |
| RKP-$\delta$-CC with $\delta = 0\%$ and $\Gamma = 60\%$ | 0.625                             | 0.828,125                         | 0.797,688,802                     | 0.767,252,604                     |
| RKP-$\delta$-CC with $\delta = 0\%$ and $\Gamma = 80\%$ | 0.906,25                          | 0.971,354,167                     | 0.976,725,26                      | 0.982,096,354                     |

Figure 6. Average of the feasibility performance of the solutions for the RKP-$\delta$-CC under implementation uncertainty for different values of $\delta$ and $\Gamma = 60\%$ for the twelve instances of the knapsack problem.

- The results suggest that a large value of $\Gamma$ produces more attractive solutions since they have more possible outcomes under the given budget than solutions for small values of $\Gamma$. Additionally, it improves the profit produced by the RKP-$\delta$.
- RKP-$\delta$-CC with $\Gamma = 80\%$ is a very attractive solution since it improves the profit of RKP-$\delta$ with little impact on the feasibility level.

In summary, the results in Table 6 and Figure 5 show that the RKP-$\delta$-CC produces results ranging from the ones in the deterministic KP with better profits but more possible outcomes exceeding the budget, to the RKP-$\delta$ with better feasibility but smaller profit. However, the RKP-$\delta$-CC allows the decision maker to choose between the two solutions using $\Gamma$ to control it.

Finally, Table 7 and Figure 6 show how the RKP-$\delta$-CC performs for different values of $\delta$ and $\Gamma = 0\%$. The results in Table 7 and Figure 6 indicate the following.

- An increment in the $\delta$ value severely impacts the feasibility level of the RKP-$\delta$-CC. This result is expected because the problem deals with feasibility relaxation; but at the same time, the number of uncertain variables that are accepted to be affected simultaneously may exceed $\Gamma$. Therefore, there exists a higher probability of exceeding the budget.
- When combining the two controls of conservatism, the profit is almost the same as for the deterministic KPs. The impact of implementation uncertainty is insignificant as $\delta$ increases.
Table 7. Average of the objective value performance of the solutions for the RKP-$\delta$-CC under implementation uncertainty for different values of $\delta$ and $\Gamma = 60\%$ for the twelve instances of the knapsack problem.

| Formulation               | Number of uncertain variables, $u$ |
|--------------------------|-----------------------------------|
|                          | $u = 3$   | $u = 5$   | $u = 7$   | $u = 9$   |
| RKP-$\delta$-CC with $\delta = 0\%$ and $\Gamma = 60\%$ | 0.625     | 0.828,125 | 0.797,688,802 | 0.767,252,604 |
| RKP-$\delta$-CC with $\delta = 1\%$ and $\Gamma = 60\%$ | 0.375     | 0.557,291,667 | 0.622,070,313 | 0.686,848,958 |
| RKP-$\delta$-CC with $\delta = 2\%$ and $\Gamma = 60\%$ | 0.281,25  | 0.531,25  | 0.535,725,911 | 0.540,201,823 |
| RKP-$\delta$-CC with $\delta = 3\%$ and $\Gamma = 60\%$ | 0.166,666,667 | 0.273,437,5 | 0.388,183,594 | 0.502,929,688 |
| RKP-$\delta$-CC with $\delta = 4\%$ and $\Gamma = 60\%$ | 0.114,583,333 | 0.273,437,5 | 0.292,724,609 | 0.312,011,719 |
| RKP-$\delta$-CC with $\delta = 5\%$ and $\Gamma = 60\%$ | 0.041,666,667 | 0.122,395,833 | 0.205,891,927 | 0.289,388,021 |

Table 8. Average of the proportion of the objective value of selected solutions for the RKP-$\delta$.

| Formulation               | Selected solution | Number of uncertain variables, $u$ |
|--------------------------|-------------------|-----------------------------------|
|                          | $u = 1$   | $u = 3$   | $u = 5$   | $u = 7$   | $u = 9$   |
| Deterministic            | 0.997,033,621 | 0.994,518,15 | 0.996,384,812 | 0.987,165,005 | 0.978,865,513 |
| RKP-$\delta$ with $\delta = 0\%$ Average | 0.990,173,046 | 0.970,353,468 | 0.948,841,825 | 0.929,545,441 | 0.909,664,756 |
| $\hat{x}_{UB}$           | 0.992,063,492 | 0.974,499,804 | 0.953,944,885 | 0.936,827,957 | 0.919,546,851 |
| $\hat{x}_{LB}$           | 0.988,247,012 | 0.964,508,58 | 0.942,658,73 | 0.921,626,984 | 0.906,749,902 |
| $\hat{x}_{D}$            | 0.988,299,532 | 0.974,499,804 | 0.946,567,863 | 0.932,206,759 | 0.915,308,151 |
| $\hat{x}_{R}$            | 0.988,299,532 | 0.974,499,804 | 0.946,567,863 | 0.932,206,759 | 0.915,308,151 |
| RKP-$\delta$ with $\delta = 1\%$ Average | 0.997,682,864 | 0.978,384,486 | 0.956,907,73 | 0.937,770,394 | 0.918,061,331 |
| $\hat{x}_{UB}$           | 1.000,980,777 | 0.984,307,572 | 0.961,494,904 | 0.944,508,449 | 0.927,227,343 |
| $\hat{x}_{LB}$           | 0.994,223,108 | 0.974,648,986 | 0.952,579,365 | 0.929,166,667 | 0.907,539,683 |
| $\hat{x}_{D}$            | 0.998,011,928 | 0.978,878,648 | 0.961,021,505 | 0.937,818,752 | 0.924,850,895 |
| $\hat{x}_{R}$            | 0.995,051,546 | 0.984,307,572 | 0.961,494,904 | 0.944,508,449 | 0.927,227,343 |

- The level of feasibility tends to increase as the level of uncertainty (the number of uncertain variables) increases, offering more attractive solutions when the decision maker knows that a high level of uncertainty is possible.

Previous results show the average performance of the robust solutions in the set $U(x)$. Section 4.2 shows how to select specific solutions within this set having certain characteristics. Consider the RKP-$\delta$ with the different levels of $\delta$ and the solutions described in Table 4. This case study computes each solution in Table 4 for each instance of the knapsack problem and the proportion of the profit of the deterministic KP, $f(x)/f^*$, for each solution. Table 8 presents the average of the proportion of the profit of each solution for the twelve instances of the knapsack problem.

These results suggest that robust solutions produce a total return within a small range, thus producing total return values with low variability, guaranteeing a stable objective solution to the decision maker.

7. Conclusions

In practice, the prescribed solutions to optimization problems may require to be changed during implementation for unforeseen reasons not considered during the optimization model development. This work presents a model for possible changes to the prescribed solution, as implementation uncertainty affects binary decision variables. The impact of implementation uncertainty on binary variables may be more significant because a change in the values in these variables inverts the prescribed decision. This article focuses on the RBIU-$\delta$ that limits the scope of the problem to the context of binary linear programs and the case where there is no information about the probability distributions describing the uncertainties. The main solution approach is a robust optimization approach with a distinct characteristic: uncertainty affects decision variables rather than model parameters. The proposed solution methodology permits the decision maker to control the level of conservatism...
associated with worst-case solutions. Additionally, the proposed method allows a reformulation of the problem to a new BLP formulation depending on deterministic variables only, thus reducing the number of decision variables of the original BLP and simplifying its solution from a practical standpoint.

A case study applies the proposed approach to the knapsack problem, assuming implementation uncertainty is present, and evaluates the profit obtained and the proportion of possible outcomes that exceed the given budget. The case study shows the proposed robust approach’s practical simplicity and benefits. Results on the knapsack problem suggest that robust solutions display beneficial properties compared to their deterministic counterparts in terms of possible outcomes resulting from uncertainty, guaranteeing the desired level of possible outcomes that satisfy the budget constraint, compared to the significant impact on the possible outcomes exceeding the budget of the deterministic solution. Furthermore, the robust solutions offer a profit close to the deterministic solution’s despite high uncertainty. Moreover, the results show that the control of feasibility relaxation and cardinality-constrained parameters reduce the impact on the profit of the robust solutions and guarantee high levels of feasibility, providing the decision maker the ability to control feasibility according to the characteristics of the problem at hand.

The robust solution set may be large, as it grows exponentially with the number of uncertain variables. Therefore, this work proposes selecting specific solutions in the robust solution set such that the prescribed (selected) solution also possesses desirable characteristics for implementation. The results of the case study suggest that the robust set may contain robust solutions with close-to-optimal performance for the deterministic version of the problem with high levels of feasibility.

This work opens opportunities for future research; for instance, the development of measures of objective degradation less conservative than the worst-case objective value. It is also of interest to solve the optimization problem over the robust set directly, i.e. without identifying the robust set as an intermediate step. Also, there is an opportunity to apply the proposed methodologies in other application contexts.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data that support the findings of this study are available from the corresponding author, Jose Ramirez-Calderon (JR), upon reasonable request.

ORCID

Jose Ramirez-Calderon http://orcid.org/0000-0002-8414-6597

References

Beale, Evelyn M. L. 1955. ”On Minimizing a Convex Function Subject to Linear Inequalities.” Journal of the Royal Statistical Society. Series B (Methodological) 17 (2): 173–184. doi: 10.1111/j.2517-6161.1955.tb00191.x.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski. 2009. Robust Optimization. Princeton, NJ: Princeton University Press.

Ben-Tal, Aharon, and Arkadi Nemirovski. 1998. “Robust Convex Optimization.” Mathematics of Operations Research 23 (4): 769–805.

Ben-Tal, Aharon, and Arkadi Nemirovski. 1999. “Robust Solutions of Uncertain Linear Programs.” Operations Research Letters 25 (1): 1–13.

Ben-Tal, Aharon, and Arkadi Nemirovski. 2000. “Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data.” Mathematical Programming 88 (3): 411–424.

Ben-Tal, Aharon, and Arkadi Nemirovski. 2002. “Robust Optimization—Methodology and Applications.” Mathematical Programming 92 (3): 453–480.
Bertsimas, Dimitris, and Melvyn Sim. 2003. “Robust Discrete Optimization and Network Flows.” Mathematical Programming 98 (1–3): 49–71.
Bertsimas, Dimitris, and Melvyn Sim. 2004. “The Price of Robustness.” Operations Research 52 (1): 35–53.
Dantzig, George B. 1955. “Linear Programming Under Uncertainty.” Management Science 1 (3–4): 197–206.
Das, Indraneel. 1997. “Nonlinear Multicriteria Optimization and Robust Optimality.” PhD diss., Rice University.
Deb, Kalyanmoy, and Himanshu Gupta. 2006. “Introducing Robustness in Multi-Objective Optimization.” Evolutionary Computation 14 (4): 463–494.
Eichfelder, Gabriele, Corinna Krüger, and Anita Schöbel. 2017. “Decision Uncertainty in Multiobjective Optimization.” Journal of Global Optimization 69 (2): 485–510.
El Ghaoui, Laurent, and Hervé Lebret. 1997. “Robust Solutions to Least-Squares Problems with Uncertain Data.” SIAM Journal on Matrix Analysis and Applications 18 (4): 1035–1064.
El Ghaoui, Laurent, François Oustry, and Hervé Lebret. 1998. “Robust Solutions to Uncertain Semidefinite Programs.” SIAM Journal on Optimization 9 (1): 33–52.
Gabrel, Virginie, Cécile Murat, and Aurélie Thiele. 2014. “Recent Advances in Robust Optimization: An Overview.” European Journal of Operational Research 235 (3): 471–483.
Jornada, Daniel, and V. Jorge Leon. 2016. “Bicriteria Robust Optimization Over the Efficient Set for Pareto Set Reduction.” European Journal of Operational Research 252 (2): 573–586. doi: 10.1016/j.ejor.2016.01.017.
Kang, Jin-Su, Tai-Yong Lee, and Dong-Yup Lee. 2012. “Robust Optimization for Engineering Design.” Engineering Optimization 44 (2): 175–194.
Kouvelis, Panos, and Gang Yu. 1997. Robust Discrete Optimization and Its Applications. Vol. 14 of the series Nonconvex Optimization and Its Applications. New York: Springer. doi: 10.1007/978-1-4757-2620-6.
Lewis, Adrian S., and C. H. Jeffrey Pang. 2009. “Lipschitz Behavior of the Robust Regularization.” SIAM Journal on Control and Optimization 48 (5): 3080–3104.
Mulvey, John M., Robert J. Vanderbei, and Stavros A Zenios. 1995. “Robust Optimization of Large-Scale Systems.” Operations Research 43 (2): 264–281.
Ramirez Calderon, Jose Ernesto. 2018. “Robust Binary Linear Programming under Implementation Uncertainty.” PhD diss. Texas A & M University.
Soyster, Allen L. 1973. “Technical Note—Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming.” Operations Research 21 (5): 1154–1157.
Wets, Roger J.-B. 1966. “Programming Under Uncertainty: The Equivalent Convex Program.” SIAM Journal on Applied Mathematics 14 (1): 89–105.
Wets, Roger J.-B. 1974. “Stochastic Programs with Fixed Recourse: The Equivalent Deterministic Program.” SIAM Review 16 (3): 309–339.
Wets, Roger J.-B. 1983. “Solving Stochastic Programs with Simple Recourse.” Stochastics: An International Journal of Probability and Stochastic Processes 10 (3–4): 219–242.