Omega 53 (2015) 124–137

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

A practical guide to robust optimization[☆]

Bram L. Gorissen[a], İhsan Yanıkoğlu[a,b,*], Dick den Hertog[a]

[a] Center for Economic Research (CentER), Tilburg University, 5000 LE Tilburg, The Netherlands
[b] Industrial Engineering Department, Özyeğin University, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 11 November 2013
Accepted 24 December 2014
Available online 3 January 2015

Keywords:
Robust optimization
Adjustable robust optimization

ABSTRACT

Robust optimization is a young and active research field that has been mainly developed in the last 15 years. Robust optimization is very useful for practice, since it is tailored to the information at hand, and it leads to computationally tractable formulations. It is therefore remarkable that real-life applications of robust optimization are still lagging behind; there is much more potential for real-life applications than has been exploited hitherto. The aim of this paper is to help practitioners to understand robust optimization and to successfully apply it in practice. We provide a brief introduction to robust optimization, and also describe important do's and don'ts for using it in practice. We use many small examples to illustrate our discussions.

1. Introduction

Real-life optimization problems often contain uncertain data. Data can be inherently stochastic/random or it can be uncertain due to errors. The reasons for data errors could be measurement/estimation errors that come from the lack of knowledge of the parameters of the mathematical model (e.g., the uncertain demand in an inventory model) or could be implementation errors that come from the physical impossibility to exactly implement a computed solution in a real-life setting. There are two approaches to deal with data uncertainty in optimization, namely robust and stochastic optimization. *Stochastic optimization* (SO) has an important assumption, i.e., the true probability distribution of uncertain data has to be known or estimated. If this condition is met and the reformulation of the uncertain optimization problem is computationally tractable, then SO is the methodology to solve the uncertain optimization problem at hand. For details on SO, we refer to Prékopa [44], Birge and Louveaux [22] and Ruszczyński and Shapiro [48], but the list of references can be easily extended.

Robust optimization (RO), on the other hand, does not assume that probability distributions are known, but instead it assumes that the uncertain data resides in the so-called uncertainty set. Additionally, basic versions of RO assume "hard" constraints, i.e., constraint violation cannot be allowed for any realization of the data in the uncertainty set. RO is popular because of its computational tractability for many classes of uncertainty sets and problem types. For a detailed overview of the RO framework, we refer to Ben-Tal et al. [9], Ben-Tal and Nemirovski [6] and Bertsimas et al. [17].

Although the first published study [49] dates back to 1970s, RO is a relatively young and active research field, and has been mainly developed in the last 15 years. There have been many publications that show the value of RO in many fields of application including finance [41], energy [19,1], supply chain [8,40], healthcare [30], engineering [5], scheduling [54], marketing [53], etc. Indeed, the RO concepts and techniques are very useful for practice, since they are tailored to the information at hand and leads to tractable formulations. It is therefore remarkable that real-life applications are still lagging behind; there is much more potential for real-life applications than has been exploited hitherto.

In this paper we give a concise description of the basics of RO, including the so-called *adjustable* RO for multi-stage optimization problems. Moreover, we extensively discuss several items that are important when applying RO, and that are often not well understood or incorrectly applied by practitioners. Several important do's and don'ts are discussed, which may help the practitioner to successfully apply RO. We use many small examples to illustrate our discussions.

The remainder of the paper is organized as follows. Section 2 gives a concise introduction to RO. Sections 3–10 discuss several important practical issues in more detail. An important ingredient of RO is the so-called *uncertainty set*, which is the set of values for the uncertain parameters that are taken into account in the robust problem. This set has to be specified by the user, and Section 3 presents several ways to construct uncertainty sets. Section 4 discusses an important technical aspect in *adjustable* RO. In practice, multi-stage problems may contain adjustable variables that are integer. Section 5 proposes a RO method that deals with

[☆] This manuscript was processed by Associate Editor Storer.
[*] Corresponding author.
E-mail addresses: b.l.gorissen@uvt.nl (B.L. Gorissen), ihsan.yanikoglu@ozyegin.edu.tr (İ. Yanıkoğlu), d.denhertog@uvt.nl (D. den Hertog).

http://dx.doi.org/10.1016/j.omega.2014.12.006
0305-0483/© 2014 Elsevier Ltd. All rights reserved.

such integer variables. Section 6 warns that the robust versions of equivalent deterministic problems are not necessarily equivalent, and gives the correct robust counterparts. Section 7 discusses that equality constraints containing uncertain parameters needs a special treatment. In RO one optimizes for the worst case scenario, i.e., for the worst case values of the uncertain parameters. Although this statement is true, the way it is formulated is often misunderstood. In Section 8, we therefore clarify how it should be formulated. It is important to test how robust the final solution is, and to compare it to, e.g., the nominal solution. Section 9 discusses how to assess the robustness performance of a given solution via a simulation study. Section 10 shows that RO applied in a folding horizon setting may yield better solutions than *adjustable* RO for multi-stage problems. Section 11 summarizes our conclusions, and indicates future research topics.

2. Introduction to robust optimization

In this section we first give a brief introduction to RO, and then we give a procedure for applying RO in practice. The scopes of the following sections are also presented in this section.

2.1. Robust optimization paradigm

For the sake of exposition, we use an uncertain linear optimization problem, but we point out that most of our discussions in this paper can be generalized for other classes of uncertain optimization problems. The "general" formulation of the uncertain linear optimization problem is as follows:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} : \mathbf{A}\mathbf{x} \le \mathbf{d}\}_{(\mathbf{c},\mathbf{A},\mathbf{d})\,\in\,\mathcal{U}}, \tag{1}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{d} \in \mathbb{R}^m$ denote the uncertain coefficients, and $\mathcal{U}$ denotes the user specified uncertainty set. The "basic" RO paradigm is based on the following three assumptions [9, p. xii]:

A.1. All decision variables $\mathbf{x} \in \mathbb{R}^n$ represent "here and now" decisions: they should get specific numerical values as a result of solving the problem before the actual data "reveals itself".

A.2. The decision maker is fully responsible for consequences of the decisions to be made when, and only when, the actual data is within the prespecified uncertainty set $\mathcal{U}$.

A.3. The constraints of the uncertain problem in question are "hard", i.e., the decision maker cannot tolerate violations of constraints when the data is in the prespecified uncertainty set $\mathcal{U}$.

In addition to the "basic" assumptions, we may assume without loss of generality that (1) the objective is certain; (2) the constraint right-hand side is certain; (3) $\mathcal{U}$ is compact and convex; and (4) the uncertainty is constraint-wise. Below, we explain the technical reasons of why these four assumptions are not restrictive.

E.1. Suppose the objective coefficients ($\mathbf{c}$) are uncertain and (say) these coefficients reside in the uncertainty set $\mathcal{C}$:

$$\min_{\mathbf{x}}\ \max_{\mathbf{c}\in\mathcal{C}}\ \{\mathbf{c}^\top\mathbf{x} : \mathbf{A}\mathbf{x} \le \mathbf{d} \quad \forall \mathbf{A} \in \mathcal{U}\}.$$

Without loss of generality we may assume that the uncertain objective of the optimization problem can be equivalently reformulated as certain [9, p. 10]:

$$\min_{\mathbf{x},\,t}\{t : \mathbf{c}^\top\mathbf{x} - t \le \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C},\ \mathbf{A}\mathbf{x} \le \mathbf{d} \quad \forall \mathbf{A} \in \mathcal{U}\},$$

using a reformulation and the additional variable $t \in \mathbb{R}$.

E.2. The second assumption is not restrictive because the uncertain right-hand side of a constraint can always be translated to an uncertain coefficient by introducing an extra variable $x_{n+1} = -1$.

E.3. The uncertainty set $\mathcal{U}$ can be replaced by its convex hull conv ($\mathcal{U}$), i.e., the smallest convex set that includes $\mathcal{U}$, because testing the feasibility of a solution with respect to $\mathcal{U}$ is equivalent to taking the supremum of the left hand side of a constraint over $\mathcal{U}$, which yields the same optimal objective value if the maximization is conv($\mathcal{U}$). For details of the formal proof and the compactness assumption, see [9, pp. 12–13].

E.4. To illustrate that robustness with respect to $\mathcal{U}$ can always be formulated constraint-wise, consider a problem with two constraints and with uncertain parameters d_1 and d_2: $x_1 + d_1 \le 0$, $x_2 + d_2 \le 0$. Let $\mathcal{U} = \{\mathbf{d} \in \mathbb{R}^2 : d_1 \ge 0,\ d_2 \ge 0,\ d_1 + d_2 \le 1\}$ be the uncertainty set. Then, $\mathcal{U}_i = [0, 1]$ is the projection of $\mathcal{U}$ on d_i. It is easy to see that robustness of the i-th constraint with respect to $\mathcal{U}$ is equivalent to robustness with respect to $\mathcal{U}_i$, i.e., the uncertainty in the problem data can be modelled *constraint-wise*. For the general proof, see [9, pp. 11–12].

For uncertain *nonlinear* optimization problems, the assumptions are also without loss of generality, except the third basic assumption [E.3].

If we assume that $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^m$ are certain, then the robust reformulation of (1) that is generally referred to as the *robust counterpart* (RC) problem is given as follows:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} : \mathbf{A}(\boldsymbol{\zeta})\mathbf{x} \le \mathbf{d} \quad \forall \boldsymbol{\zeta} \in \mathcal{Z}\}, \tag{2}$$

where $\mathcal{Z} \subset \mathbb{R}^L$ denotes the user specified *primitive* uncertainty set. A solution $\mathbf{x} \in \mathbb{R}^n$ is called *robust feasible* if it satisfies the uncertain constraints $[\mathbf{A}(\boldsymbol{\zeta})\mathbf{x} \le \mathbf{d}]$ for all realizations of $\boldsymbol{\zeta} \in \mathcal{Z}$.

As it is mentioned above, and explained in [E.4], we may focus on a single constraint, since the uncertainty is constraint-wise in RO. A single constraint taken out of (2) can be modeled as follows:

$$(\mathbf{a} + \mathbf{P}\boldsymbol{\zeta})^\top\mathbf{x} \le d \quad \forall \boldsymbol{\zeta} \in \mathcal{Z}. \tag{3}$$

In the left-hand side of (3), we use a factor model to formulate a single constraint of (2) as an affine function $\mathbf{a} + \mathbf{P}\boldsymbol{\zeta}$ of the *primitive* uncertain parameter $\boldsymbol{\zeta} \in \mathcal{Z}$, where $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{P} \in \mathbb{R}^{n\times L}$. One of the most famous example of such a factor model is the 3-factor model of Fama and French [29], which models different type of assets as linear functions of a limited number of uncertain economic factors. To point out, the dimension of the general uncertain parameter $\mathbf{P}\boldsymbol{\zeta}$ is often much higher than that of the primitive uncertain parameter $\boldsymbol{\zeta}$ (i.e., $n \gg L$).

2.2. Solving the robust counterpart

Notice that (3) contains infinitely many constraints due to the *for all* ($\forall$) quantifier imposed by the worst case formulation, i.e., it seems intractable in its current form. There are two ways to deal with this. The first way is to apply robust reformulation techniques to exclude the *for all* ($\forall$) quantifier. If deriving such a robust reformulation is not possible, then the second way is to apply the adversarial approach. In this section, we describe the details of these two approaches.

We start with the first approach, which consists of three steps. The result will be a computationally tractable RC of (3), which contains a finite number of constraints. Note that this reformulation technique is one of the main techniques in RO [17].