Robust Optimization of System Design

Evgeny Shindin, Odellia Boni, Michael Masin*

IBM Research, Mount Carmel, Haifa 31905, Israel

Abstract

The data of real-world optimization problems are usually uncertain, that is especially true for early stages of system design. Data uncertainty can significantly affect the quality of the nominal solution. Robust Optimization (RO) methodology uses chance and robust constraints to generate a robust solution immunized against the effect of data uncertainty. RO methodology can be applied to any generic optimization problem where one can separate uncertain numerical data from the problem's structure. Since 2000, the RO area is witnessing a burst of research activity in both theory and applications. However, RO could lead to over-conservative requirements, resulting in typical-case bad solutions or even empty solution spaces. This drawback of the classical RO methodology can be overcome by distinguishing between real decision variables and so-called state variables. While the first type should satisfy the chance or robust constraints and their value cannot depend on a specific realization of the uncertain data, the state variables are adjustable (i.e., their value can depend on the specific realization of the uncertain data), since most of the constraints defining state variables merely “calculate” their exact value, and hence are always satisfied. In this paper we summarize how adjustable RO approach can be applied to a general uncertain linear optimization problem. Then, using an allocation example we demonstrate how this approach can be integrated in the design optimization process and its impact on the optimal system design.

© 2014 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the University of Southern California.

Keywords: Robust Optimization; State Variables; System Design; Adjustable Robust Counterpart; Uncertainty Set

* Corresponding author. Tel.: +972-48281369; fax: +972-4-829-6115.

E-mail address: michaelm@il.ibm.com
1. Introduction

The data of real-world optimization problems more often than not are uncertain – not known exactly at the time the problem is being solved. The reasons for data uncertainty include, among others, measurement and estimation errors coming from the impossibility to precisely measure/estimate the data entries representing the characteristics of physical systems/technological processes/environmental conditions, etc. In addition, implementation errors coming from the impossibility to implement a solution exactly as it is computed, could also be modeled as data uncertainty. In real-world applications of optimization, one cannot ignore cases where a small uncertainty in the data can make the nominal optimal solution completely meaningless. The Robust Optimization (RO) offers a methodology capable of detecting cases when data uncertainty can heavily affect the quality of the nominal solution, and generate a robust solution immunized against the effect of data uncertainty. The goal of RO is to find a robust optimal solution, i.e. find values for decision variables which are feasible for all possible values of uncertain parameters, while optimizing the uncertain objective. By itself, the RO methodology can be applied to any generic optimization problem where uncertain numerical data belonging to a given uncertainty set could be separated from the certain problem structure (i.e., goals, constraints, and decision variables).

The origins of RO date back to the establishment of modern decision theory in the 1950s and the use of the worst case analysis. The paradigm of RO per se, goes back to A.L. Soyster\(^1\) who was the first to consider, as early as in 1973, what now is called Robust Linear Programming. In two subsequent decades there were only two publications on the subject\(^2,3\). The activity in the area was revived circa 1997, independently and essentially simultaneously, in the frameworks of both Integer Programming\(^4\) and Convex Programming\(^5,6,7,8\). Since 2000, the RO area is witnessing a burst of research activity in both theory and applications, with numerous researchers involved worldwide.

The standard way to deal with Robust Optimization problem is to find a computationally tractable certain optimization problem called (Approximated) Robust Counterpart (RC), which solution is feasible for the original problem's robust constraints, meets its chance constraints with corresponding probabilities and is (approximately) optimal for its objective. The RO methodology is constraint-wise, i.e. it is applied sequentially per problem constraint. Different modeling techniques and principles are typically applied to robust inequality constraints, robust equality constraints and to chance constraints in order to transform the uncertain problem into its robust counterpart. It should be noted that these special techniques are generally not known to non-experts.

When trying to apply the RO methodology to real life problems we face several challenges, since the classical robust optimization is suitable only where the following set of assumptions hold:

- All decision variables represent “here and now” decisions; they should be assigned specific numerical values as a result of solving the problem before the actual data “reveals itself”.
- The decision maker is fully responsible for consequences of the decisions to be made when, and only when, the actual data is within the pre-specified uncertainty set \(U\).
- The constraints are hard, meaning that we cannot tolerate violations of constraints, even small ones, when the data is in \(U\).

However, real life problems often involve state variables, which are not “real” decision variables, thus the first assumption can be relaxed. These variables are called adjustable variables and the constraints containing these variables can be treated differently. The resulting counterpart is called Adjustable Robust Counterpart\(^9\) (ARC). In addition, while the original uncertain design problem can be mixed integer linear problem (MILP), its (approximated) Robust Counterpart could be non-linear. Unfortunately, most of existing optimization solvers are not suitable to solve non-linear problems efficiently. Hence, it is important to find linear formulations or approximations of the uncertain problems counterparts. Another challenge is the formulation of uncertainty sets. In literature, the uncertainty set \(U\) is usually considered to be independent of the problem structure and is given in an explicit form. However, in real-life problems the definition of \(U\) is not given explicitly and may depend on the value of some of the decision variables. Moreover, real uncertainties can depend on decisions not always explicitly described as decision variables in the original uncertain optimization problem.

Thus, in spite of existing classical techniques, transformation of an uncertain real-life problem to a tractable approximation of its robust counterpart may be a hard and complex process. Consider the following uncertain MILP problem:
min \( c^T x + \gamma^T s + g \) \hfill (1a)

s.t. \( a_i^T x + d_i^T s + b_i \leq 0 \) \( i = 1, \ldots, I_1 \) \hfill (1b)

\( a_i^T x + d_i^T s + b_i = 0 \) \( i = I_1 + 1, \ldots, I_1 + I_2 \) \hfill (1c)

\( P(a_i^T x + \Delta_i^T s > \beta_i) \leq \epsilon_k \) \( k = 1, \ldots, K \) \hfill (1d)

\( x, s \geq 0 \) \hfill (1e)

\( (c, g, a_i, b_i, a_k, \beta_k) \in U \) \hfill (1f)

where \( a_i, c, a_k \) are \( J_1 \) uncertain parameters vectors, \( b, g, \beta_k \) are uncertain scalar parameters, \( d, \gamma, \Delta_k \) are \( J_2 \) certain parameters vector, \( \epsilon_k \) are probabilities for the chance constraints, \( x \) is \( J_1 \) vector of decision variables which could contain continuous or discrete components and \( s \) is \( J_2 \) vector of state variables. We denote by \( \Psi \) set of all uncertain parameters of Problem (1). The values of these parameters are affected by uncertainty set \( U \). We assume \( U \) of \( L \) independent uncertainty sources:

\[
U = U^0 + \sum_{\ell=1}^L \epsilon^\ell U^\ell = \left[ c^0, g^0, a_i^0, b_i^0, a_k^0, \beta_k^0 \right] + \sum_{\ell=1}^L \epsilon^\ell \left[ c^\ell, g^\ell, a_i^\ell, b_i^\ell, a_k^\ell, \beta_k^\ell \right], \quad |\epsilon^\ell| \leq 1
\]

where \( U^0 = \left[ c^0, g^0, a_i^0, b_i^0, a_k^0, \beta_k^0 \right] \) is a set of nominal values (one for each uncertain parameter) and \( U^\ell = \left[ c^\ell, g^\ell, a_i^\ell, b_i^\ell, a_k^\ell, \beta_k^\ell \right] \) is a set of basic shifts due to uncertainty source \( \ell \). Given any parameter \( \psi \in \Psi \) we denote its nominal value by \( \psi^0 \) and its basic shift by \( \psi^\ell \). The formulation of Problem (1) is suitable to represent a large class of practical optimization problems, including system and system of system design with mapping functional requirements to physical architecture, resource allocation and scheduling, reliability and timing calculations, power and data flow, etc. One can see that (1) is a complex problem mixing robust inequality constraints, robust equality constraints, chance constraints and adjustable variables. Such problems could have a computationally intractable RC. Fortunately, (1) has following nice properties, making its RC tractable:

- The problem is affine both in space of decision variables and uncertain parameters.
- The coefficients of adjustable variables \( s \) are certain, i.e. the Problem (1) is a problem with fixed recourse.

In Section 2, we present steps for reformulating Problem (1) with decision dependent uncertainty set, in order to obtain an equivalent formulation of the problem following the same form (1) and having a decision independent uncertainty set (2). In Section 3, we present a methodology integrating many techniques from Ben-Tal et al.\(^9\) to transform a complex uncertain MILP (1) containing robust inequality, robust equality and chance constraints with adjustable (state) variables and uncertainty set (2) to a MILP approximation of its RC. We discuss the complexity of the obtained approximation in terms of number of variables and constraints. In Section 4, we show an example demonstrating how to combine the steps described in Sections 2 and 3 in order to build a MILP approximated RC of a real problem subjected to decision-dependent uncertainty. Finally, we shortly summarize in Section 5.

### 2. Representing uncertainty sources

The basic shifts of the uncertain parameters are often represented as deviation percentage rates \( \sigma^\ell_\psi \) from their nominal value: \( \psi^\ell = \sigma^\ell_\psi \psi^0 \) where \( \sigma^\ell_\psi \) is decision independent or dependent (e.g., bus latency is 5±20% msec). In the following, we analyze the modifications to the model (1) and the new uncertainty set (2) to support uncertainties of this specific form while maintaining linearity of constraints. To define the new uncertainty set, we need to define all basic shift sets \( U^\ell \) caused by uncertainty sources \( \ell = 1, \ldots, L \). For each \( \ell \) we go over all uncertain parameters \( \psi \in \Psi \) and define corresponding basic shift value \( \psi^\ell \). In case when \( \psi \) not affected by source \( \ell \) we set \( \psi^\ell = 0 \).

The deviation rates may be decision-independent or depend on the values of decisions presented in the model, e.g. bus latency rate may depend on the bus type. For the first case, where \( \sigma^\ell_\psi = \delta^\ell_\psi \) is a constant, we set the
corresponding basic shift value as $\psi' = \delta'_c \psi^0$ without any modifications to the original problem. When dealing with decision-dependent uncertainties, we need to add new parameters $\psi'$ to the parameters set $\Psi$, define corresponding nominal values and basic shifts, and reformulate left hand side (LHS) of Problem (1) constraints. We may also need to add certain constraints and variables. We summarize these modifications in Table 1.

The first step is to recognize uncertainties depending on decisions, which are not included in the set of decision variables, i.e., implied decisions. If such uncertainties exist, the corresponding implied decision variables and the constraints defining the relations between the existing and these implied variables must be added to the Problem (1). We assume a simple model of rate dependency: $\sigma_v^d = \delta_v^d x_i$, where each dependent deviation rate is a linear function of a single non-adjustable decision variable. The assumption is reasonable and very generic since $x_i$ could be an implied “complex” decision. Next, uncertainty rates are classified by the types of parameters influenced by this uncertainty (free parameters or variables coefficients). For a dependent rate affecting a free parameter (the first case in Table 1), an addition of a single term to the relevant constraint and of a single parameter to the uncertainty set is sufficient. However, for an uncertainty rate which depends on variable $x_j$ and affects the coefficient of variable $x_j$ at some constraint, linearity of the constraint can be maintained only if at least one of the variables ($x_j$ or $x_i$) has a finite number of different values (either $x_j$ or $x_i$), $x_m$ is the other variable ($x_i$ or $x_j$, respectively), $N$ is the number of different values of $x_m$, $\eta^m_0, \eta^m$ reflect the values of $x_m$, and by $W$ we denote a large number.

<table>
<thead>
<tr>
<th>Affected parameter</th>
<th>Reformulation</th>
<th>New parameters values</th>
<th>New parameter set</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi = g, b, \beta_k$</td>
<td>LHS$' = LHS + \tilde{\psi}x_i$</td>
<td>$\tilde{\psi}^0 := 0, \tilde{\psi}' := \delta'_c$</td>
<td>$\Psi' \subseteq \Psi \cup \tilde{\psi}$</td>
</tr>
<tr>
<td>$\psi = c, a_{i,j}, \alpha_{k,j}$</td>
<td>LHS$' = LHS + \tilde{\psi}x_m + \sum_{m=1}^{N-1} \psi^m_n x_m^n</td>
<td>\tilde{\psi}^0 := 0, \tilde{\psi}' := \psi^0 \delta^m_c \eta^m_n$</td>
<td>$\Psi' \subseteq \bigcup_{n=0}^{N-1} \tilde{\psi}_n \cup \Psi$</td>
</tr>
<tr>
<td>$\psi = \eta^m_0, \eta^m$</td>
<td>$x_m = \eta^m_0 + \sum_{n=1}^{N-1} \eta^m_n \rho^m_n$, $\rho^m_n \in {0,1}$</td>
<td>$\sum_{n=1}^{N-1} \rho^m_n \leq 1$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$y^m_n \leq W \rho^m_n$, $x_m - y^m_n \leq W(1 - \rho^m_n)$</td>
<td>$y^m_n \geq -W \rho^m_n$, $x_m - y^m_n \geq -W(1 - \rho^m_n)$</td>
<td></td>
</tr>
</tbody>
</table>

### 3. Creating approximated robust counterpart

We start creating ARC by splitting the set of decision variables to non-adjustable ($x$) and adjustable ($s$) variables. The process is straightforward and depends on the variable context. All real decisions are non-adjustable, such as what components are used; all temporary variables, created to calculate some problem metrics, e.g., state variables, slacks, and so on, are adjustable. Then, we analyze constraints: each constraint containing adjustable variables and/or uncertain parameters is a robust constraint otherwise it is a certain constraint and should not be transformed. Similarly, we decide if the objective is certain or should be transformed.

Chance constraints could be linearly approximated by applying budgeted uncertainty introduced by Bertsimas and Sim. Ben-Tal et al. have shown that, there exists a less conservative RC of chance constraints, but budgeted uncertainty has one major advantage – it could be represented by a system of linear constraints. Moreover, recently developed variable budgeted uncertainty could successfully reduce conservativeness of budgeted RC preserving linearity of its representation. For clarity, in this paper we use the original simplest (and most conservative) approximation involving the following constants depending on the total number of uncertainty sources $L$: 

[Note: The table and mathematical expressions are simplified for clarity and may not reflect the exact presentation in the original document.]
\[ \lambda_k = \sqrt{-2L \ln\left(1 - \varepsilon_k\right)}. \]  

(3)

Table 2 – Transformation of Uncertain Problem into its Adjustable Robust Counterpart for uncertainty set (2)

<table>
<thead>
<tr>
<th>Original problem objects</th>
<th>Reformulation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Adjustable variable</strong></td>
<td><strong>Objective (1a)</strong></td>
</tr>
<tr>
<td>( s_j )</td>
<td>( u_0' ) \quad -u_0' \leq \left[ c' \right] x + \gamma' q' + g' \leq u_0' )</td>
</tr>
<tr>
<td><strong>new variables</strong></td>
<td>( \min \quad t \left{ \left[c'\right] x + \gamma' p + \sum_{i=1}^{L} u_i' + g' \leq t \right} )</td>
</tr>
<tr>
<td><strong>new constraints</strong></td>
<td></td>
</tr>
<tr>
<td><strong>new variables</strong></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1 – Concise Modeling of the Task Allocation Example

The certain approximation for various types of uncertain objects (variables, objective, and constraints) is shown in Table 1. One can see that each uncertain object of the original problem is transformed into a set of linear constraints and additional variables: \( L+1 \) variables replace a single adjustable variable, \( L \) additional variables and \( 2 \times L+1 \) inequality constraints replace an uncertain objective, \( L \) additional variables and \( 2 \times L+1 \) inequality constraints replace each robust inequality, \( L+1 \) equality constraints replace each robust equality, \( 3 \times L+2 \) additional variables and \( 4 \times L+2 \) inequalities and \( L \) equalities replace each chance constraint and \( L \) additional variables and \( 2 \times L+1 \)
inequalities replace each non-negativity constraint. Thus, the approximated RC size may greatly increase, especially in case of large $L$.

4. Tasks allocation example

Let us consider the following design problem of a system that can contain up to $N$ processors and must complete $J$ tasks before a predefined threshold of $\max\text{Makespan}$. Each task consists of $O$ operations processed sequentially from operation 1 to operation $O$. Operation 3 for each task must be done immediately after Operation 2. There are $T$ types of processors, each of a different cost and different types of operations it can perform. Each operation has a known processing time, dependent on task and operation, but independent of processor’s place or type. We decide what processors to buy and what operations and in what sequence each processor performs. Design goal is to minimize the total system cost. Concise Modeling of the problem is shown in Fig. 1. This problem is common for AUTOSAR and IMA architectures and could be formulated as the following MILP problem:

$$\min \ total\ Cost = \sum_{n=1}^{N} \sum_{t=1}^{T} ecu_{n,t} \cdot \text{cost}_t,$$

$$\text{s.t.} \quad \sum_{t=1}^{T} t \cdot ecu_{n,t} \leq \sum_{t=1}^{T} t \cdot ecu_{n-1,t} \quad \forall n=1,\ldots,N \quad \text{(4a)}$$

$$ct_{j,o} \geq ct_{j,o-1} + pt_{j,o} \quad \forall j=1,\ldots,J, \quad o=2,\ldots,O, o \neq 3 \quad \text{(4b)}$$

$$ct_{j,o} = ct_{j,o-1} + pt_{j,o} \quad \forall j=1,\ldots,J, \quad o=3 \quad \text{(4c)}$$

$$\text{makespan} \geq ct_{j,o} \quad \forall j=1,\ldots,J \quad \text{(4d)}$$

$$\text{makespan} \leq \max\text{Makespan} \quad \text{(4e)}$$

$$ct_{j,o_2} \geq ct_{j,o_1} + pt_{j_2,o_2} - W(3 - o2e_{j_1,o_1,n} - o2e_{j_2,o_2,n} - x_{j_1,o_1,j_2,o_2}) \quad \forall n,j_1 \neq j_2,o_1,o_2 \quad \text{(4f)}$$

$$x_{j_1,o_1,j_2,o_2} = 0 \quad \forall j_1 = j_2,o_1,o_2 \quad \text{(4g)}$$

$$x_{j_1,o_1,j_2,o_2} \leq W \quad \forall j_1 \neq j_2,o_1,o_2 \quad \text{(4h)}$$

$$x_{j_1,o_1,j_2,o_1} \leq 1 \quad \forall j_1 \neq j_2,o_1,o_2 \quad \text{(4i)}$$

$$x_{j_1,o_2,j_2,o_1} \leq 1 - (2 - o2e_{j_1,o_1,n} - o2e_{j_2,o_2,n}) \quad \forall n,j_1 \neq j_2,o_1,o_2 \quad \text{(4j)}$$

$$\sum_{o=1}^{O} o2e_{j,o,n} = 1 \quad \forall j,o \quad \text{(4k)}$$

$$o2e_{j,o,n} \leq \sum_{t=1}^{T} ecu_{n,t} \quad \forall j,o,n \quad \text{(4l)}$$

$$ecu_{n,t}, x_{j_1,o_1,j_2,o_2}, o2e_{j,o,n} \in [0,1], ct_{j,o}, total\ Cost, makespan \geq 0 \quad \text{(4m)}$$

where the problem's parameters are $pt_{j,o}$ - processing time for operation $o$ of task $j$, $ecu_t$ - a set of operations which could be performed by processor of type $t$, and $W$ which is a large number. The problem's variables are $ecu_{n,t}$ - a decision to place processor of type $t$ to place $n$, $o2e_{j,o,n}$ - a decision to perform operation $o$ of task $j$ on processor $n$, $x_{j_1,o_1,j_2,o_2}$ - a decision to perform operation $o1$ of task $j_1$ before operation $o2$ of task $j_2$, $ct_{j,o}$ - the completion time of operation $o$ of task $j$, and makespan is the maximum completion time for all tasks.

Solving certain Problem (4a) with sample data the optimal solution has three processors and total cost of 1300. However, system analysis identifies the following independent uncertainty sources affecting processing times:

- global uncertainty could affect all processing times simultaneously at rate $\delta_1$,
- per operation uncertainties at rates $\delta_{j,o}, o=1,\ldots,O$,
- per processor uncertainties at rates $\delta_{n,o}, n=1,\ldots,N$,
- per processor type uncertainties at rates $\delta_{t,O,N+1}, t=1,\ldots,T$.

In this case, implementing the optimal nominal solution results in violation of the makespan threshold for most realizations of the uncertain processing times. On the other hand, the optimal solution of the classical RC (that assumes worst possible processing times and does not take adjustability into account) consists of four processors and increases the total cost to 1800. In the following we apply the adjustable RO methodology described above.
First, when analyzing uncertainty sources one can see that the last uncertainty source depends on our decision: which operation of which task we set to which processor type. However, there are no such decision variables in the original problem formulation. Hence, new decision variables \( e_{\text{cu}, \alpha, t} \) are created. These variables depend on existing decision variables \( o_{2 \tau, j, \alpha, n} \) and require additional constraints:

\[
o_{2 \tau, j, \alpha, n} \geq o_{2 \tau, j, \alpha, n} + e_{\text{cu}, \alpha, t} - 1, \quad \sum_{t=1}^{T} o_{2 \tau, j, \alpha, t} = 1 \quad (4o)
\]

The uncertainty sources analysis reveals that \( L = 1 + O + N + T \). Applying modifications according to Table 1 we obtain the following set of basic shifts (note that the first two sources are decision-independent):

\[
U' = \begin{cases}
\bigcup_{\tau \neq \alpha} p_{\tau, j, \alpha} = \delta_{\tau, j, \alpha} p_{\tau, j, 0} & \ell = 1 \\
\bigcup_{\tau \neq \alpha} p_{\tau, j, \alpha} = \delta_{\tau+1, j, \alpha} p_{\tau, j, 0} & 1 < \ell \leq O + 1 \\
\bigcup_{\tau \neq \alpha} b_{\tau, j, \alpha} = \theta b_{\tau, j, 0, n} = \ell - (O + 1) & O + 1 < \ell \leq 1 + O + N \\
\bigcup_{\tau \neq \alpha} a_{\tau, j, \alpha} = \delta_{\tau+O, N, \tau} p_{\tau, j, 0, t} = \ell - (1 + O + N) & 1 + O + N < \ell \leq 1 + O + N + T
\end{cases}
\]

(5)

where \( a_{\tau, j, \alpha}, b_{\tau, j, 0, n} \) are new parameters with nominal values \( a_{\tau, j, \alpha}^0 = b_{\tau, j, 0, n} = 0 \).

Reformulations from Table 1 also affect Constraints (4d, e, h) that become as follows:

\[
c_{j, \alpha, t} = c_{j, \alpha, 1} + p_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} a_{\tau, j, \alpha} 2_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} b_{\tau, j, \alpha} o_{2 \tau, j, \alpha, n} \quad \forall j = 1...J, \alpha = 2...O, \alpha \neq 3 \\
c_{j, \alpha} = c_{j, \alpha, 1} + p_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} a_{\tau, j, \alpha} 2_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} b_{\tau, j, \alpha} o_{2 \tau, j, \alpha, n} \quad \forall j = 1...J, \alpha = 3 \\
c_{j, \alpha, t} = c_{j, \alpha, 1} + p_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} a_{\tau, j, \alpha} 2_{\tau, j, \alpha} + \sum_{\tau = 1}^{\ell} b_{\tau, j, \alpha} o_{2 \tau, j, \alpha, n} - W(3-o_{2 \tau, j, \alpha, n} - o_{2 \tau, j, \alpha, n} x_{\tau, j, \alpha, n}) \quad \forall n, j \neq j, \alpha, 0, 2
\]

(6)

Next, we start building the ARC. One can see that \( e_{\text{cu}, \alpha, t}, x_{\tau, j, \alpha, 0, 2}, o_{2 \tau, j, \alpha, n} \) are real decision variables, while \( c_{j, \alpha, t} \) and \( \text{makespan} \) are state, i.e., adjustable, variables. The objective (4a) and the Constraints (4b,c,i-m,o) do not contain uncertain parameters or adjustable variables, and, hence, are certain and remain unchanged. Inequality Constraints (4f-g) and non-negativity Constraints (4n) contain adjustable variables but do not depend on any uncertain parameters; hence, their ARC can be reformulated according to the relevant rows in Table 2:

\[
-u_{\tau, j, \alpha} \leq u_{\tau, j, \alpha} - \text{makespan}' \leq u_{\tau, j, \alpha} \quad \forall \ell, j \quad \text{makespan} \geq \text{ctp}_{j, \alpha} + \sum_{\alpha \neq 1} u_{\alpha, j, \alpha} \quad \forall j
\]

\[
-c_{\tau, j, \alpha} \leq c_{\tau, j, \alpha} \leq c_{\tau, j, \alpha} \quad \forall \ell, j, \alpha
\]

\[
\text{ctp}_{j, \alpha} - \sum_{\alpha \neq 1} c_{\tau, j, \alpha} \geq 0 \quad \forall j, \alpha
\]

\[
-\text{makespan}' \leq \text{makespan}' \leq \text{makespan}' \quad \forall \ell
\]

\[
\text{makespan} - \sum_{\alpha \neq 1} \text{makespan}' \geq 0
\]

\[
\text{makespan} + \sum_{\alpha \neq 1} \text{makespan}' \leq \max\text{Makespan}
\]

where \( \text{ctp}_{j, \alpha}, c_{\tau, j, \alpha}, \text{makespan}, \text{makespan}' \) are new decision variables replacing the adjustable variables, and \( u_{\tau, j, \alpha}, c_{\tau, j, \alpha}, \text{makespan}, \text{makespan}' \) are additional decision variables.

Finally, we modify inequality Constraints (6) (that replaced Constraints (4d, e, h)) containing the uncertain processing times \( p_{\tau, j, \alpha} \) according to Table 2:

\[
-u_{\tau, j, \alpha} \leq c_{\tau, j, \alpha} + p_{\tau, j, \alpha} + \sum_{\alpha \neq 1} a_{\tau, j, \alpha} 2_{\tau, j, \alpha} + \sum_{\alpha \neq 1} b_{\tau, j, \alpha} o_{2 \tau, j, \alpha, n} - c_{\tau, j, \alpha} \leq u_{\tau, j, \alpha} \quad \forall \ell, j, \alpha \neq 3
\]

\[
\text{ctp}_{j, \alpha} \geq \text{ctp}_{j, \alpha} + p_{\tau, j, \alpha} + \sum_{\alpha \neq 1} u_{\tau, j, \alpha} \quad \forall j, \alpha \neq 3, \alpha > 1
\]
\[ c_{\ell, j, o} = c_{\ell, j, o-1} + p_{f, j, o} + \sum_{i=1}^{T} a_{f, j, o,i} o^{2} t_{j, o,i} + \sum_{n=1}^{N} b_{f, j, o,n} o^{2} e_{j, o,n} - c_{\ell, j, o} \quad \forall \ell, j, o = 3 \]

\[ c_{f, j, o} = c_{f, j, o-1} + p_{f, j, o} \quad \forall f, j, o = 3 \]

\[ -u_{n, j, 1, j, 2, 0, 2} \leq c_{1, j, 1, 0} + p_{1, j, 2, o} + \sum_{i=1}^{T} a_{j, 2, o,i} o^{2} t_{j, 2, o,i} + \sum_{n=1}^{N} b_{j, 2, o,n} o^{2} e_{j, 2, o,n} - \\
- W (3 - o^{2} e_{j, 0, 0, 1, 0, 2} - o^{2} e_{j, 2, 0, 1, 0, 2} - x_{j, 0, 1, 2, 0, 2}) - c_{1, j, 1, 0} \leq u_{n, j, 1, j, 2, 0, 2} \quad \forall \ell, n, j, \neq j_{2}, o_{1}, o_{2} \]

\[ c_{f, j, 0, 2} \geq c_{f, j, 1, 0} + p_{f, j, 2, o} - W (3 - o^{2} e_{j, 0, 0, 1, 0, 2} - o^{2} e_{j, 2, 0, 1, 0, 2} - x_{j, 0, 1, 2, 0, 2}) + \sum_{i=1}^{T} u_{2, j, 1, 0, 1, 2, 0, 2} \]

where \( u_{f, j, o, 2} \) are additional decision variables.

In our example, solving the ARC with the sample uncertainty rates, we obtain an optimal solution with three processors of total cost 1400. This solution is robust to all predefined processing time uncertainties.

5. Summary

In this paper we summarized and demonstrated all steps required to build a tractable robust counterpart of an uncertain system design problem with state variables and different kinds of uncertainty. In the provided example of system design, we showed that the adjustable RO approach can considerably improve the worst time analysis results and ensure compliance with product requirements with only an incremental increase in cost.

Acknowledgement

This work was supported in part by European Commission for funding the Large-scale integrating project (IP) proposal under the ICT Call 7 (FP7-ICT-2011-7) 'Designing for Adaptability and evolutioN in System of systems Engineering (DANSE)', contract 287716

References