Solving advanced multi-objective robust designs by means of multiple objective evolutionary algorithms (MOEA): A reliability application

Daniel E. Salazar A., Claudio M. Rocco S.

*División de Computación Evolutiva (CEANI), Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI), Universidad de Las Palmas de Gran Canaria, Canary Islands, Spain

bUniversidad Central de Venezuela, Facultad de Ingeniería, Caracas, Venezuela

Received 15 October 2005; received in revised form 14 March 2006; accepted 23 March 2006

Abstract

This paper extends the approach proposed by the second author in [Rocco et al. Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application. In: Tarantola S, Saltelli A, editors. SAMO 2001: Methodological advances and useful applications of sensitivity analysis. Reliab Eng Syst Saf 2003;79(2):149-59 [special issue]] to obtain a robust system design. The approach based on the use of evolutionary algorithms and interval arithmetic finds the maximum-volume inner box (MIB) or the maximal ranges of variation for each variable that preserve pre-specified design/performance requirements. The original single-objective formulation considers the definition of a MIB around a specified centroid (case 1), or around an unspecified centroid (case 2). In this paper, both cases were successfully modified and solved as multiple-objective (MO) problems, showing the advantages of MO formulations in a design-selection decision framework. Special attention is devoted to the unspecified centre MO problem where the computational efficiency could be a critical issue. In that sense, a new procedure based on the “percentage representation” is proposed. This approach reduces drastically the computational burden, extending the possibilities of use of robust design.

Keywords: MOEA; Multiple-objective optimisation; Percentage representation; Robust design

1. Introduction

Robust design is defined as the process of identifying by which values design variables should be set to in order to minimise the output variability (see e.g. Ref. [2]). This task is often accomplished by an iterative procedure. In this approach different design settings are evaluated by quantifying the effects of the uncertainty that these settings cause in the design’s output.

Such experimental procedure might be performed either in a physical way (e.g. in a laboratory) or emulating numerically the behaviour of the design variables by means of mathematical models and a suitable characterisation of the uncertainty associated to the design variables, i.e. by their probability distribution functions.

As a result, this class of procedure yields a set point at which the system shows its highest level of insensitivity to the input variations (e.g. see Ref. [2]). This approach can be conjugated with the optimisation of some design features, producing simultaneously an optimal and more-robust search. Some examples of methodologies that attempt to find both robustness and optimality are described in Refs. [3,4]. The main concept, over which the whole robust design procedure is based, consists of minimising the variability.

Nevertheless there is another concept of robust design. It is defined as the maximum size of the deviation from the target design that can be tolerated, whereby the product...
still meets all requirements (see Ref. [5]). This idea can be interpreted as the output’s variability does not necessarily have to be minimised but rather that it be bounded. Therefore any design setting that lies inside such bound is an acceptable setting, and the one with the largest tolerance to the input uncertainty is considered as the robust design. This is the approach adopted by Refs. [1,6].

Let us exemplify the above-mentioned concept: consider the reliability of a system modelled as \( R_S = f(R_1, R_2, \ldots, R_n) \) where \( R_i \) is the reliability of component \( i \). Now suppose the designer is willing to accept a predefined system reliability requirement, let us say \( 0.990 \leq R_S \leq 0.999 \). In this case any setting of component reliabilities that produces \( 0.990 \leq R_S \leq 0.999 \) is considered as a feasible design. On the other hand, the robustness is associated with the maximum size of the deviation from the variables setting which is consistent with the predefined requirement.

In Ref. [1] the described concept is formulated as an optimisation problem where the size of the deviation is maximised subject to the designer’s requirements (domain constraints and objective goals). The mathematical problem is then solved by a hybrid heuristic approach where an evolution-strategy technique serves as the optimisation tool, while interval arithmetic is used as a checking technique that guarantees the feasibility of the design, providing strictly bounds with only one “interval” evaluation. The approach presented in Ref. [1] is able to efficiently solve different problems with high quality.

However, there are situations in which the decision makers (DM) need to optimise two or more objectives, for example cost and reliability. Since the model in Ref. [1] is based on a single-objective (SO) formulation, any designer or DM must solve several problems by varying a group of constraints to obtain a set of alternatives from which to choose the final solution.

In this paper we extend the previous work and propose a multiple-objective (MO) formulation, which is able to generate a set of alternatives, based on two or more conflicting objectives, from which the DM can choose, a posteriori, the preferred one. This formulation allows optimising some objectives while assuring robustness.

The rest of this paper is organised as follows: the remainder of this section describes the mathematical formulation of the original robust-design approach proposed in Ref. [1]. Section 2 describes the multiple-objective formulation, the evolutionary multiple-objective optimisation procedure, an efficient-representation scheme called “percentage representation” and other implementation details. Section 3 is devoted to the computational examples, and finally Section 4 presents the final remarks.

1.1. Problem description

As an example, consider the life-support system in a space capsule [7] shown in Fig. 1. The symbolic reliability expression of the system, where \( R_i \) is the reliability of component \( i (i = 1, 2, 3, 4) \), is given by \([8]\)

\[
R_S = 1 - R_3(\bar{R}_4R_2)^2 - \bar{R}_3(1 - R_3(1 - \bar{R}_1\bar{R}_4))^2.
\]

The user can be interested to know the maximum-possible deviations for each component in Fig. 1 consistent with a predefined aspiration level of performance, e.g., \( 0.990 \leq R_S \leq 0.999 \).

Suppose the area shown in Fig. 2a is the feasible zone \( F \) for a generic design with variables \( R \) and \( R_n \). Within the feasible zone any pair \((R, R_n)\) satisfies the specifications. An exact description of the feasible-solution set (FSS) (Fig. 2a) is in general not simple, since it may be a very complex set. Moreover, FSS could be limited by non-linear functions. For this reason, approximate descriptions are often looked for, using simply shaped sets like boxes or ellipsoids containing outer bounding (Fig. 2b) or contained in inner bounding (Figs. 2c and 2d) the set of interest \([9,10]\). In particular, minimum-volume outer box (Fig. 2b) and maximum-volume inner box (MIB) (Figs. 2c and 2d) are of interest. In this study only the MIB determination is considered.

The maximum ranges of possible variations of the feasible values are the sizes (along co-ordinate axis) of the axis-aligned box of minimum volume containing FSS. To obtain the MIB it is required that all the points inside
the generated box satisfy the constraints. Therefore, the mathematical formulation is [1]:

Let $B$ be the box defined by

$$B := \{x, \ C \in \mathbb{R}^n | x_j \in [x_j^1, x_j^u] \} \quad C_i = (x_i^u + x_i^l)/2.$$

Two cases are analysed:
(1) MIB, centre specified

The idea is to produce a symmetrical MIB using a given point $C$ as symmetry centre.

(2) MIB, centre unspecified

In this case, the centre of co-ordinates $C$ is unknown and it is considered as an additional variable. The goal is to produce a symmetrical MIB around $C$.

In both cases, the objective is to maximise the inner volume:

1. Centre specified : $\max_x \prod_{i=1}^{n} |x_j - C_i| \quad s.t. : x \in F \cap B$.

2. Centre unspecified : $\max_x \prod_{i=1}^{n} |x_j - C_i| \quad s.t. : x \in F \cap B$.

The objective functions represent a quantity that is proportional to the MIB hyper-volume. Note that $x$ represents a vertex of the optimal MIB. From here, the range of each variable is easily determined.

As previously mentioned, there are situations in which DM need to optimise two or more objectives. For example, in the previous example, DM could be interested in how to balance reliability and cost: is it better to select a design with higher reliability and higher cost or a design with lower cost sacrificing reliability? [11].

Using a single-objective (SO) formulation, DM must solve several problems by varying a group of constraints to obtain a group of alternatives from which to choose the final solution. On the contrary, a multiple-objective (MO) approach allows determining directly the Pareto set of alternatives from which the DM can choose, a posteriori, the preferred one.

In Ref. [6] the authors propose an approach based on the use of multiple-objective evolutionary algorithms (MOEA), which is able to handle constraints and solve the centre-specified problem effectively. However the approach suggested for solving the centre-unspecified problem involves considerable computational effort.

In this paper an effective approach based on the “percentage representation” (PC) is proposed. This representation was introduced in Ref. [12] to overcome the difficulties that arise during the evolutionary search when dependences exist between the current values of some variables and the upper bound of the search space of other variables that belong to the same chromosome. In the percentage representation, the search space is transformed by representing those dependent variables as percentage and decodifying them in a suitable way.

2. Robust-design multiple-objective formulation

A MO optimisation problem consists of optimising a vector of functions:

$$\text{Opt}(F(x) = f_1(x), f_2(x), \ldots, f_k(x))$$

s.t. : $g_j(x) \leq 0, j = 1, 2, \ldots, q.$

$h_j(x) = 0, j = 1, 2, \ldots, r(g + r = m)$

where $x \in X$ is the solution vector, or vector of decision variables, and $X$ is the feasible domain.

Following the above definition, the robust-design problems 1 and 2 previously presented (Eqs. (1) and (2)) can be formulated as a multiple-objective optimisation problem (MOP), transforming one or more constraints into objectives. For example, using the maximal cost $C_{S_{\text{max}}}$ of a MIB as an objective yields:

1. Centre specified : $\max_x \prod_{i=1}^{n} |x_j - C_i| \land \min C_{S_{\text{max}}}$

2. Centre unspecified : $\max_x \prod_{i=1}^{n} |x_j - C_i| \land \min C_{S_{\text{max}}}$.

Notice that even when the two objectives considered in these MOP types are MIB and cost, the MOP approach is general and can be used for any type and number of objectives (for example MIB and weight). The selection of such objectives clearly relies on the problem under study and the DM criteria.

Note also that the concept of optimality must be adapted for a MOP, since the single-objective definition cannot be directly applied. In consequence, different concepts of optimality have been proposed, the Pareto optimality, being the most extensively used, in which the order of the solutions is built according to the following definitions [13]. Without loss of generality, in terms of minimisation we have:

**Definition 1**: Pareto optimal: A solution vector $x^* \in X$ is Pareto Optimal solution of a $k$-objective optimisation problem iff $\neg \exists x \in X : f_i(x) \leq f_i(x^*) \land f(x) \neq f(x^*); i = \{1, 2, \ldots, k\}$. These solutions are also called true Pareto solutions.

**Definition 2**: Pareto dominance: A solution $x^1$ dominates $x^2$, denoted $x^1 \succ x^2$, iff $f_i(x^1) \leq f_i(x^2) \land \exists j : f_j(x^1) > f_j(x^2); i, j = \{1, 2, \ldots, k\}$. If there are no solutions which dominates $x^1$, then $x^1$ is non-dominated.

**Definition 3**: Pareto approximation set: A set of non-dominated solutions $\{x^i\} = \exists x : x \succ x^i; x^i, x \in D \subseteq X$ is said to be a Pareto set. If $D = X$, then the set is a true Pareto Set.

**Definition 4**: Pareto front: the set of vectors in the objective space that are image of a Pareto set, i.e. $\{F(x^i)\} = \exists x : x \succ x^i; x^i, x \in D \subseteq X$ is a Pareto front or a
Pareto approximating front. If $D = X$, then the set constitutes a true Pareto front.

The solving procedure of a MOP can be accomplished by means of two different approaches. The first formulates and solves the problem as a MOP, based on the specified criteria (e.g. maximise reliability and minimise cost), whereas the second approach transforms the original MOP into several single-objective optimisation problems (SOP) to be solved sequentially [14,15].

In this research we consider only the former approach, i.e., we solve the robust design MOP directly using MOEA. This family of evolutionary algorithms are tailored to deal with MOP as well as handling constraints. The approach does not guarantee the determination of the exact Pareto frontier nor does any heuristic approach for that matter. Nevertheless, an important number of comparisons (e.g. [16–18]) performed in evolutionary multicriteria optimisation on benchmark problems have shown that results obtained using different instances of MOEA are very close to the exact solution. A description of the MOEA approach is presented in the following subsections. Some issues like MOEA efficiency comparisons, tuning or complexity, are not considered in the following exposition.

### 2.1. Multiple-objective evolutionary algorithms

In the evolutionary multicriteria optimisation field the term multiple-objective evolutionary algorithms (MOEA) refers to a group of evolutionary algorithms tailored to deal with MOP. This group of algorithms conjugates the basic concepts of dominance described in the later section with the general characteristics of evolutionary algorithms. Therefore, MOEA are able to deal with non-continuous, non-convex and/or non-linear spaces, as well as problems whose objective functions are not explicitly known (e.g. the output of Monte Carlo simulation runs).

Since the first recognised MOEA (Schaffer’s VEGA [1984] [19]), the development of MOEA has successfully evolved, producing better and more-efficient algorithms. The existing MOEA might be classified into two groups [20], according to its characteristics and efficiency. On the one hand there is a first group known as “first-generation” which includes all the early MOEA (aggregating functions [21], VEGA [19], MOGA [21], NPGA [22], NSGA [23]). On the other hand there is a second group named “second-generation MOEA,” which comprises of very efficient optimisers like SPEA2 [18], PESA-II [24], NSGA-II [16] and micro-GA [21] among others. Basically, this last group differs because it incorporates a mechanism of adaptation or fitness function in terms of dominance and an elitist archive which stores the best solutions found so far during the evolutionary process.

#### 2.2. Nondominated sorting genetic algorithm II

The nondominated sorting genetic algorithm (NSGA-II) is a well known and extensively used algorithm based on its predecessor NSGA. It was formulated by Deb et al. [16] as a fast and very-efficient MOEA that incorporates the features mentioned earlier, i.e. an elitist archive and a rule for adaptation assignment that takes into account both the rank and the distance of each solution regarding others.

Fig. 3 describes a way of implementing NSGA-II. First the elitist population or archive $P_A$ is initialised with random values whereas the population $P$ is set to empty. Then, the individuals of both populations are put together and the adaptation is assigned in terms of dominance, according to Ref. [16]. The best $N$ individuals are assigned to the archive, where the individuals that serve as parents in the recombination phase (crossover/mutation) are selected from. The offspring compose the new population and the process is repeated until the finalisation criterion is reached.

### 2.3. Constraint-handling techniques

One of the main issues in evolutionary computation is how to guide the search towards the feasible region in the presence of constraints. The existing approaches can be classified in the following groups: (1) Penalisation techniques, (2) Repairing techniques, (3) Separation techniques, and (4) Hybrid techniques. For a review see Ref. [25].

The experiments reported in this section are based on the approaches proposed by Deb et al. [16], which could be implemented in the NSGA-II as follows: calculate the normalised sum of constraint violations for all the individuals belonging to population and the archive ($P \cup P_A$). Classify the individuals according to the overall constraints violation: when comparing two individuals; if the overall violation of both of them is zero, apply the ordinary ranking assignment, otherwise the individual with the lower (or null) overall violation dominates the other one. The rest of the algorithm remains equal.
The integration of this technique to any MOEA promotes feasibility over optimality. Thus the search is guided toward the feasible region. Once it is reached, feasible individuals are sorted according to the particular fashion established by the MOEA instance. Moreover, as the reader may notice, the absence of penalisation parameters saves the effort of tuning.

2.4. The percentage representation

The percentage representation was introduced first in Ref. [12] to deal with a floating bound scheduling problem. The problem is characterised by a dependency between the current value of some variables and the bounds of others. Ref. [12] to deal with a floating bound scheduling problem. The problem is characterised by a dependency between the current value of some variables and the bounds of others.

As an example, assume that a chromosome is composed of two variables \( v_1 \) and \( v_2 \), where \( v_1 \in [a,b] \) and \( v_2 \in [c,d] \) for some fixed values \( a, b \) and \( c \). Now suppose we have two individuals \( A = \{v_1^1, v_2^1\} \), \( v_1^1 \in [a,b] \wedge v_2^1 \in [c,d] \) and \( B = \{v_1^2, v_2^2\} \), \( v_1^2 \in [a,b] \wedge v_2^2 \in [c,d] \). Thus, in the valid case when \( v_1^1 < v_2^1 \leq v_2^2 \) the potential offspring of a crossover of A and B are \( \{v_1^1, v_2^2\} \) which is feasible, and \( B = \{v_1^1, v_2^2\} \) which is unfeasible since it violates \( v_2 \in [c,d] \).

Clearly, to be able to explore the whole domain of the problem without generating unfeasible individuals, \( v_1 \) must be fixed before assigning values to \( v_2 \). This leads to a "double-loop" configuration, where the value of \( v_1 \) is controlled by an external loop whereas the value of \( v_2 \) is handled by a nested loop and the search space is conformed by a set of disjoint subspaces. This dependency problem appeared first in Ref. [26].

In the percentage representation, the values of dependent variables are considered as percentage of the variables they depend on. In this way, the search space is defined as an augmented domain where only feasible individuals are generated and the recombination operators can be applied without restrictions. The next subsection describes how the percentage representation was implemented in this work.

2.5. Implementation

NSGA-II is implemented following the pseudo-code presented in Fig. 3. Our implementation is suitable both for integer chromosomes, real chromosomes and for mixed chromosomes (for more details see Ref. [27]). For the particular problem studied here, only real variables were needed.

The recombination mechanism is one-point crossover, adapted for a mixed chromosome. For real variables, the crossover is performed as a linear combination while the mutation operation is performed like Gaussian mutation of type \( R_{new} = R_{old} + N(0,\sigma^2) \).

The decision variables of the \( n \)-dimensional problem are the centroid \( C \) and the lower vertex \( x^l \) where \( C_i, x_i^l \in [C_{i,\text{min}}, C_{i,\text{max}}] \), \( 0 \leq C_{i,\text{min}}, C_{i,\text{max}} \leq 1 \). For each \( C \), the following constraints stand: the lower vertex must verify \( C_{i,\text{min}} \leq x_i^l \leq C_i \) whereas the upper vertex, which can be calculated as \( x_i^u = 2C_i - x_i^l \) due to the symmetry condition imposed, is restricted to \( C_i \leq x_i^l \leq C_i^{\text{max}} \).

A simple MOEA approach can be used directly to analyse the centre-specified case [6], nevertheless, the centre-unspecified case requires more attention. As a matter of fact, there exists a dependency between the bounds of \( x_i^l \) and the value of \( C \). As we mentioned above, at least two strategies can be employed to solve the problem with MOEA. The first one consists of a double-loop configuration, where the external loop controls the value of \( C \) and the nested loop finds the best vertex for each prefixed \( C \). The process is repeated iteratively for different values of \( C \), up to a representative number of different centroids have been visited.

The other alternative transforms the search space by means of a percentage representation, codifying each individual as a group of \( n \) pairs \( \{C_i, \%x_i^{\text{max}}\} \) where \( \%x_i^{\text{max}} \) represents a percentage of the maximal distance between \( C_i \) and its limits. Therefore, the mathematical relation to determine the value of \( x_i^l \) is:

\[
\%x_i^{\text{max}} = \frac{|C_i - x_i^l|}{x_i^{\text{max}}} = \frac{|C_i - x_i^l|}{\min(|C_i - C_i^{\text{min}}|, |C_i - C_i^{\text{max}}|)}.
\]

Note that in principle, \( x_i^l \) could be either the lower or the upper vertex. Thus, simply introducing a decision during the decodification, we obtain the value of \( x_i^l \) as:

\[
x_i^l = \begin{cases} 
C_i = \%x_i^{\text{max}} \cdot (C_i - C_i^{\text{min}}) & \text{if } C_i - C_i^{\text{min}} < C_i^{\text{max}} - C_i \\
C_i - \%x_i^{\text{max}} \cdot (C_i^{\text{max}} - C_i) & \text{otherwise}.
\end{cases}
\]

Since the vertex and centroid relationship is relative, the search space is unified and only feasible individuals can be produced when the recombination operators are applied.

The following section presents an application example that demonstrates the advantage of this type of representation in the problem addressed.

3. Computational example

3.1. Centre-specified robust design MOP

The proposed approach is applied to define a robust design for the system [7] shown in Fig. 1. Both SOP and MOP optimisation results are presented.

The problem is to obtain the ranges for each \( R_i \) such as \( 0.99 \leq R_S \leq 1 \), subject to: \( 0.80 \leq R_i \leq 1 \), with centre of each interval as: \( C_i = 0.90 \), \( i = 1,2,3,4 \).

Table 1 shows the MIB obtained in Ref. [28] for the centre specified case (Eq. (1)). These ranges produce an output \( R_S \) belonging to: [0.9900, 0.9999].

Normally the design problem seeks to constrain a cost function (in arbitrary cost units), such as \( C_S = 2 \sum K_iR_i^{l} \)

In this case \( K_i \) is a proportionality constant and \( x_i \) the exponential factor (usually less than one) that relates the cost of each component and its reliability. This means that \( K_i \) is the cost of component \( i \) when \( R_i = 1 \) and \( K_iR_i^{l} \) is the
reduced cost when \( R_i < 1 \) \[29\]. For example in \[7\]: 
\[ K_1 = 100, \quad K_2 = 100, \quad K_3 = 100, \quad K_4 = 150; \quad \alpha_i = 0.6 \quad \forall i. \]

Using the above values and the ranges shown in Table 1 for \( R_i \), the range for \( C_S \) is \([791.369485, 896.123195]\).

As previously mentioned, the approach can consider several constraints simultaneously. For example, we can solve the problem to obtain the ranges for each \( R_i \) such as \( 0.99 \leq R_S \leq 1 \) and \( C_{\min} \leq C_S \leq C_{\max} \). In this case, if we define: \( C_{\min} = 800 \) and \( C_{\max} = 870 \), then the new solution is shown in Table 2.

The MO formulation is obtained converting the cost constraint into an objective function, yielding a centre-specified MOP (Eq. (3)). Thus, the following problem is a bi-objective problem with centre-specified \( (C_i = 0.90) \). In those cases where the original SOP considers several constraints, higher-dimensional MO formulations are possible. Nevertheless, the bi-objective formulation is the only case suitable for an easy visual inspection; therefore it is appropriate as an introductory example.

Fig. 4 shows the Pareto front determined by NSGA-II in Ref. [6] for two experiments using 12550 evaluations of the objective function. Note that the approximation Pareto fronts obtained are practically equals; thus for the sake of readability, in the subsequent figures only one experiment is presented. From this figure, it is also clearly visible that the solution achieved during the SO optimisation, as expected, shows the same approximation quality.

Fig. 5 shows a complete picture of the situation. The minimum and maximum cost as a function of the MIB volume is presented for the solutions obtained earlier. As expected, as long as the range of variation on reliability components is wider, and therefore the MIB is bigger, the range for cost becomes more uncertain. In consequence, DM should define how much uncertainty can be allowed for the system. In other words, the robustness of the system has a price, which is directly related with the maximal cost of the chosen MIB.

In that sense, Fig. 6 shows the average component tolerance in terms of the MIB. For instance, if the DM defines a maximum allowed cost of 870 units, then the maximum average component tolerance would be in the range from 0% to almost 10%. The average component tolerance was calculated as the arithmetic mean of the tolerances of each variable while the variable tolerance is the percentage of allowed variation around the centroid that is defined by the MIB.

### Table 1
Robust design using CES and IA approach: A single constraint on \( R_S \) \[28\]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Starting interval</th>
<th>Final interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_1 )</td>
<td>[0.80, 1.00]</td>
<td>[0.8001, 0.9998]</td>
</tr>
<tr>
<td>( R_2 )</td>
<td>[0.80, 1.00]</td>
<td>[0.8220, 0.9779]</td>
</tr>
<tr>
<td>( R_3 )</td>
<td>[0.80, 1.00]</td>
<td>[0.8032, 0.9967]</td>
</tr>
<tr>
<td>( R_4 )</td>
<td>[0.80, 1.00]</td>
<td>[0.8652, 0.9347]</td>
</tr>
</tbody>
</table>

Final intervals produce: \( R_S = [0.995628, 0.999835] \) and \( C_S = [820.8062, 868.4570] \).

### Table 2
Robust design using CES and IA approach: Constraints on both \( R_S \) and \( C_S \) \[28\]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Starting interval</th>
<th>Final interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_1 )</td>
<td>[0.80, 1.00]</td>
<td>[0.854327, 0.945673]</td>
</tr>
<tr>
<td>( R_2 )</td>
<td>[0.80, 1.00]</td>
<td>[0.853171, 0.946829]</td>
</tr>
<tr>
<td>( R_3 )</td>
<td>[0.80, 1.00]</td>
<td>[0.846696, 0.953304]</td>
</tr>
<tr>
<td>( R_4 )</td>
<td>[0.80, 1.00]</td>
<td>[0.870334, 0.929666]</td>
</tr>
</tbody>
</table>

Fig. 4. Two sets of trade-off approximations between MIB and \( C_{\max}^{\text{max}}[6] \).
The above results show the advantages of the MO formulation for the centre-specified formulation.

3.2. Centre-unspecified robust design MOP

To analyse the centre-unspecified MOP case (Eq. (4)), first we will illustrate the double-loop approach. In this case five sets of solutions were generated with predefined centroids. Results are shown in Fig. 7.

It is interesting to observe that the two upper curves (centroids $C_i = 0.91$ and $C_i = 0.92$) are completely dominated by the curve obtained with $C_i = 0.90$. This fact shows that a high value for centroid is not necessarily the best option, since the closeness to the constraint $R_i \leq 1$ limits the range of variation of the components reliability values and consequently reduces the MIB. Likewise, a similar mechanism reduces the extension of the lower curves (centroids $C_i = 0.86$, $C_i = 0.88$ and $C_i = 0.89$), since lower centroids are nearer the system constraint ($R_s \geq 0.99$).

Fig. 8 shows the non-dominated front for the set of obtained solutions obtained earlier. The dotted lines correspond to the dominated sectors of the curves, whereas the solid lines are the non-dominated sections. The non-dominated front in this case is formed for efficient solutions with different centroids.

This formulation is evidently better than the one used for the specified-centre case since it allows finding even better solutions. However, this formulation requires more computational effort to handle the problem of dependency between vertices and centroids. Note that this formulation consists of a time-consuming search. In fact, to generate each curve we performed 12 550 evaluations of the objective function, which yields a total of 75 300
evaluations for only 6 curves. Clearly the number of evaluations to generate each curve can be reduced, but it entails investigating the trade-off between quality of the approximation and number of evaluations, and since the result of this task depends hardly on the problem addressed, it does not sound as a good practice if one wants to have a generic methodology.

Moreover, the double-loop structure applied here is restricted to those cases where all of the values of the centroids are equals. Nevertheless, to perform a representative search, more combinations of the values of $C_i$ must be considered.

Fig. 9 shows the results obtained using the “percentage representation,” where each $R_i$ is not fixed in advance. Note that the Pareto front is enlarged, thus providing bigger MIB. However the points that produce bigger MIB correspond to some $R_i$ as low as 0.4, situation that could not be tolerated in a system. These lower-reliability values are assigned to the elements whose cost is cheaper.

To avoid this situation of lower-component reliability, an additional constraint is included that limits the lower value or $R_i$. As an example, two runs were performed, the first one with $R_i \in [0.80,1]$ and the second one with $R_i \in [0.72,1]$.

Fig. 10 shows the results for this case. Note that the Pareto fronts associated with the constrained $R_i$ are now closer to the Pareto front obtained with the double-loop approach, but with less computational effort. Only 12 550 evaluations were performed to obtain each curve, i.e., the sixth part of the evaluations performed with the double loop. On the other hand, in both cases the results obtained with the double loop as well as with the approach described in Ref. [28] are dominated by the curves generated with the percentage representation.
The above results show that the proposed approach for solving robust-design MOP provides the analyst/DM with a useful tool. Particularly, the percentage representation formulation reduces the computational burden and facilitates solving the centre-unspecified case in a very efficient way.

4. Conclusions

This paper analyses two MO formulations to obtain robust system designs. The MO formulations extend the possibilities of use of the robust-design approach, providing the DM with a wider horizon of non-dominated alternatives.

The multiple-objective evolutionary algorithm approach employed to solve the MOP formulations provides an excellent way to approximate the Pareto frontier, for both the centre-specified and the centre-unspecified cases. In addition, the proposed approach based on the percentage representation is able to effectively manage the dependency between vertices and centroids for centre-unspecified problems. Indeed the approach remarkably improves the efficiency of the MO centre-unspecified solution technique since it requires less computational effort than other techniques.
Acknowledgements

The authors are grateful to the anonymous referees for their comments and valuable suggestions on the earlier version of this paper.

References


