# Interval Robust Multi-Objective Evolutionary Algorithm

G. L. Soares, *IEEE member*, F. G. Guimarães, C. A. Maia, J. A. Vasconcelos, L. Jaulin

***Abstract*—Uncertainties are commonly present in optimization systems, and when they are considered in the design stage, the problem usually is called a robust optimization problem. Robust optimization problems can be treated as noisy optimization problems, as worst case minimization problems, or by considering the mean and standard deviation values of the objective and constraint functions. The worst case scenario is preferred when the effects of the uncertainties on the nominal solution are critical to the application under consideration. Based on this worst case scenario, we developed the [I]RMOEA (Interval Robust Multi-Objective Evolutionary Algorithm), a hybrid method that combines interval analysis techniques to deal with the uncertainties in a deterministic way and a multi-objective evolutionary algorithm. We introduce [I]RMOEA and illustrate it on three robust test functions based on the ZDT problems. The results show that [I]RMOEA is an adequate way of tackling robust optimization problems with evolutionary techniques taking advantage of the interval analysis framework.**

***Index Terms*—genetic algorithms, evolutionary algorithms, robust multi-objective optimization, robust Pareto front, robust test functions, interval analysis.**

## I. Introduction

Real world problems are usually characterized by several conflicting objectives. Solving these multi-objective problems consists in finding an estimate of the Pareto front whose solutions are as close as possible to the true Pareto front (usually unknown) and, in most cases, as diverse as possible. Many methods were proposed to achieve an estimate of the Pareto front with these desirable characteristics, such as the NSGAII [1], the PAES [2] and the SPEA [3]. Following the optimization process, the designer needs to choose a final solution to be implemented. If the uncertainties cause significant changes in the nominal solutions, the designer may use some approach based on sensitivity analysis in order to select the best solution according to the robustness requirements. Another possibility to handle uncertainties is by directly inserting them in the model, hence characterizing a RMOOP (Robust Multi-Objective Optimization Problem), which must be solved by an adequate optimization method to find the robust Pareto front. With an estimate of the robust Pareto front in hand, the designer can proceed to the decision-making phase to select the best robust solution.

The nature of the uncertainties clearly depends on the problem under analysis. Jin and Branke [4] mentioned noise, robustness, fitness approximation and time-varying fitness functions as the sources of uncertainties, and they presented a specific robust formulation for each case. From a methodological point of view, Calafiore and Campi [5], discussed some aspects of the worst case design and the probabilistic approach to solve robust convex optimization problems. While the probabilistic approaches utilize a fixed amount of samples to model the imprecision, the deterministic worst case methods compute the complete influence of uncertainties. Therefore, in a situation that involves critical conditions such that the possibility of bringing about misfortune or loss is real, the probabilistic methods should be avoided. In this context, we have decided for the worst case philosophy using Interval Analysis techniques to guarantee that the influence of all uncertainties will be considered in the optimization process.

Interval analysis was proposed by Moore [6] to deal with rounding errors in numerical computation and it has been successfully used to treat uncertainties in many real applications, see Jaulin [7] for instance. In our approach, the uncertainties are handled as additional parameters of the fitness function, which are isolated from the search domain, therefore it can be bounded and easily represented by intervals. Computing uncertainties as interval fitness functions has some advantages. First, the complete uncertainty space can be computed using only its bounds. Second, the image of any variable computed under uncertainties is guaranteed to be enclosed by an inclusion function, which is a special interval function explained later. Finally, if each interval parameter occur at most one in the interval fitness function, the worst case of a particular solution is obtained with only one calculus. Common disadvantages related to intervals are their wider interval results (nevertheless, the true solution is guaranteed to be inside the interval) and the additional computational effort of the interval methodologies. Hansen and Walster [8] refuted these criticisms and enhanced the advantages of the interval approaches such as the resolution of certain problems that cannot be solved by non-interval methods and the reliability of their results. In special, this last characteristic has motivated us to use the interval methodology to compute the worst case.

In the field of robust optimization, the majority of papers address mono-objective problems [9] or multi-objective approaches to robust mono-objective problems [10], [11]. Considering the multi-objective case, most papers use the probabilistic framework within MOEAs (Multi-Objective Evolutionary Algorithms) [12], [13]. A starting point to MOEAs beginners is Deb [1], where many MOEAs in the literature are discussed in detail. The reasons to use MOEAs generally involve the following features: a) their ability in the exploitation and exploration of the promising regions; b) their

G. L. Soares is with Pontifícia Universidade Católica (PUC Minas), Brazil (gsoares@pucminas.br); F. G. Guimarães is with the Department of Computer Science, Universidade Federal de Ouro Preto, UFOP, Brazil; C. A. Maia and J. A. Vasconcelos are with the Department of Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Brazil; and L. Jaulin is with L'École Nationale Supérieure d'Ingénieurs (ENSIETA), France.

adaptability to new formulations, purposes and environments; c) their independence on specific information about the problem, such as convexity and differentiability. Considering the advantages above, we have chosen a specific group of MOEAs, the MOGAs (Multi-Objective Genetic Algorithms), as the main structure for our search engine. MOGAs are based on the principles of evolution by natural selection and thus they work with populations of candidate solutions. As a consequence, a good estimate of the robust Pareto front can be found in a single run.

Considering the discussions above, the main contribution of this paper is to introduce interval methods in the computation of the worst-case values of the fitness functions in MOEAs. For the sake of simplicity but without loss of generality, we consider only minimization problems. The proposed [I]RMOEA is an evolutionary multi-objective technique to solve an RMOOP, considering the worst case philosophy, and using interval methods to compute the effect of uncertainties on the fitness functions. In this sense, the proposed [I]RMOEA can be considered a kind of hybrid algorithm, combining usual evolutionary operators to perform the search and deterministic interval methods to deal with the uncertainties.

This work is organized as follows. Section 2 establishes the background necessary to the further discussion. Section 4 presents the robust problem to be solved. Section 4 explains the [I]RMOEA. Section 5 validates the approach using three robust test problems based on the ZDT test suit. Section 6 presents the conclusions.

## II. Background

### A. Robust concepts

The meaning of the term robust depends on the context. For example, in biology, Kitano [14] defines robustness as the maintenance of specific functionalities of the system against environmental perturbations. In software engineering, De Vale and Koopman [15] used robustness as synonymous to fault tolerance in software. In engineering design, Du and Chen [16] considered robustness an issue related to choosing the better model by making tradeoffs between the mean and variance attributes. Usually in engineering applications, robustness means reliability of the system when, occasionally, failures happen or in the presence of the uncertainties. Even if the formal meaning changes, intuitively, the term robustness brings to mind characteristics such as strength, flexibility and adaptability.

In the optimization context, Parkinson et al. [17] present the worst case design approach. Similarly, Kouvelis and Yu [18] define the robust optimization problem as the minimization of the maximal deviation from individual criterion best values, or in other words as a minmax problem. If the robust solutions are unavailable or hard to be obtained, another one with good robust characteristics must be selected. In this sense, Roy [19] defines robustness analysis as a process to find robust conclusions that are perfectly robust, approximately robust and pseudo-robust. In Dias and Clímaco [20], robust conclusions are divided in absolute robust, relative binary robust and relative unary robust conclusions. Deb and Gupta [12] defined robust solutions as Type I and II, employing a particular formula to compute the mean that uses the image values of the perturbations in the vicinity of the solutions.

Uncertainties can be difficult to compute and it is not possible to predict their intensity. This fact motivates some researchers to focus their attention on probabilistic methods or use the classifications described above in their algorithms. In our point of view, considering the context of robust optimization, the robust solutions are those that have the best performance for the worst case of the uncertainty. Thus, by observing a given degree of accuracy, a solution is declared robust or not.

### B. Multi-Objective Genetic Algorithms

The goal of a MOGA designed to solve a multi-objective problem is to find an estimate set containing samples that are as close as possible to the true Pareto front, which is not known *a priori*, and having as much diversity as possible. The design of any MOEA has to take these two goals into account, which are known as the convergence and diversity properties of the algorithm [21].

Current MOGA's designed to find samples of the Pareto front in a multi-objective optimization problem follow the general structure presented in the Algorithm 1.

```
Data:
$k \leftarrow 0$;
$P_k \leftarrow$ Initialize population($n_{pop}$) ; /* size $n_{pop}$ */
$A_k \leftarrow \emptyset$ ;                        /* Archive set */
$O_k \leftarrow \emptyset$ ;                          /* Offspring */
begin
    while not stop criterion do
        $F_k \leftarrow$ Assign fitness($P_k$);
        $O_k \leftarrow$ Reproduction($A_k, P_k, F_k$);
        $P_{k+1} \leftarrow$ Next population($A_k, P_k, O_k$);
        $A_{k+1} \leftarrow$ Update archive($A_k, P_k$);
        $k \leftarrow k + 1$;
    end
end
```

**Algorithm 1**: Generic structure of a MOGA

The reproduction step comprises traditional selection, crossover and mutation operators. As discussed in [22], the use of an archive population is now a common strategy adopted in many MOEA's. The archive population stores the estimate set of the Pareto front and is the outcome of the algorithm. See [1] for more details on MOGA's and see [22] for a historical overview of the field.

### C. Interval Analysis

The [I]RMOEA computes the worst case performance of each solution by considering the uncertainty parameters as intervals. So, some definitions of Interval Analysis are necessary.

Denoting the real interval domain by $\mathbb{IR}$, a real, closed and connected interval $[z] \in \mathbb{IR}$ is defined as:

$$[\underline{z}, \overline{z}] = \{z \in \mathbb{R} \mid \underline{z} \leq z \leq \overline{z}\},$$

being $\underline{z}$ and $\overline{z}$, the lower and upper bounds of $[z]$. An interval vector (or box) $[\mathbf{z}] \in \mathbb{IR}^n$ is defined as the Cartesian product of $n$ intervals:

$$[\underline{\mathbf{z}}, \overline{\mathbf{z}}] = [z_1] \times [z_2] \times \ldots \times [z_n],$$

where $\underline{\mathbf{z}}$ and $\overline{\mathbf{z}}$ are the lower and the upper bounds of $[\mathbf{z}]$. Usually, the width of the boxes determines its precision. Then, if $[\mathbf{z}] \in \mathbb{IR}^n$, the width $w([\mathbf{z}])$ is given by:

$$w([\mathbf{z}]) = \max(\overline{z}_i - \underline{z}_i), \ \ i \in \{1, \ldots, n\}.$$

Using $\diamond$ to denote a binary operator that represents the four basic operations of the classic arithmetic $+$, $-$, $*$ and $/$, the interval arithmetic operations are defined as:

$$[\mathbf{z}] \diamond [\mathbf{w}] = \{\mathbf{z} \diamond \mathbf{w} \mid \mathbf{z} \in [\mathbf{z}], \ \mathbf{w} \in [\mathbf{w}]\}.$$

An operation that divides a box $[\mathbf{z}]$ in two symmetric boxes $[\mathbf{z}]_1$ and $[\mathbf{z}]_2$ is called a regular bisection. The non-overlapping boxes resulting from the regular bisection of $[\mathbf{z}]$ are:

$$\begin{aligned} [\mathbf{z}]_1 &= [z_1] \times \ldots \times [\underline{z}_j, \frac{\underline{z}_j + \overline{z}_j}{2}] \times \ldots \times [z_n], \\ [\mathbf{z}]_2 &= [z_1] \times \ldots \times [\frac{\underline{z}_j + \overline{z}_j}{2}, \overline{z}_j] \times \ldots \times [z_n], \end{aligned}$$

where $j$ is the index for the first component of maximum width, i.e., $j = \min\{i \mid w([z_i]) = w([\mathbf{z}])\}$. The union of non-overlapping boxes $[\mathbf{z}]_1$ and $[\mathbf{z}]_2$ is denoted subpaving of $[\mathbf{z}]$.

Interval functions are functions whose arguments and results are intervals. A special class of interval functions is composed by the inclusion functions. Given a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, any function $[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}^m$ that satisfies the condition:

$$\forall [\mathbf{z}] \in \mathbb{IR}^n, \ \mathbf{f}([\mathbf{z}]) \subset [\mathbf{f}]([\mathbf{z}])$$

is considered an inclusion function for $\mathbf{f}$. When the interval function involves more than one occurrence of the same interval variable, pessimistic intervals (wider intervals) are obtained. This phenomena is named dependency. For instance, consider function $f(x) = (x^2 + 2x - 1)/x$ and three possible inclusion functions:$[f]([x]) = ([x]^2 + 2[x] - 1)/[x]$, $[f]'([x]) = ([x]^2 - 1)/[x] + 2$ and $[f]''([x]) = [x] - 1/[x] - 2$. When the inclusion functions are evaluated at the interval $[1, 2]$, we have:

$$\begin{aligned} [f]([1,2]) &= ([1,2]^2 + 2[1,2] - 1)/[1,2] = [1, 7]; \\ [f]'([1,2]) &= ([1,2]^2 - 1)/[1,2] + 2 = [2, 5]; \\ [f]''([1,2]) &= [1,2] - 1/[1,2] + 2 = [2, 3.5]. \end{aligned}$$

All the results are correct but the interval provided by $[f]''([x])$ is sharper (or more precise) than the others. One of the most important tasks in Interval Analysis is to create sharp inclusion functions that accelerate the convergence of the algorithms. In order to decrease the pessimism (to obtain narrower intervals), it is necessary to minimize the occurrences of interval variables in the inclusion function expression or to treat the interval variable as a subpaving. In the previous example, consider the following non-overlapping boxes $[1, 1.5]$ and $[1.5, 2]$ obtained after one bisection in $[1, 2]$. The result is the union of all inclusion function sub-results, thus:

$$\begin{aligned} [f]([1,2]) &= [f]([1,1.5]) \cup [f]([1.5,2]) = [1.3, 4.7]; \\ [f]'([1,2]) &= [f]'([1,1.5]) \cup [f]'([1.5,2]) = [2, 4]; \\ [f]''([1,2]) &= [f]''([1,1.5]) \cup [f]''([1.5,2]) = [2, 3.5]. \end{aligned}$$

If we bisect once more $[1, 1.5]$ and $[1.5, 2]$, we will see the trend: $[f]$ and $[f]'$ converging to $[f]''$. The function $[f]''$ represents the simplest expression to represent the initial function $f$. The following theorem in Jaulin et al. [7] (p.30) states that: if an inclusion function $[f]$ is expressed as a finite composition of the operators (+,-,*,/) and the elementary functions (sin, cos, exp, ...) and each interval variable occurs only once in the formal expression, then $[f]$ has minimal width. For additional information on the quality of inclusion functions, see Ratschek and Rokne [23], and Chabert and Jaulin [24] for instance.

## III. Robust Multi-Objective Problem

We have chosen the robust formulation based on the worst case philosophy, presented as follows. Considering the design variables $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n$ and the uncertainty parameter $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^k$, the vector of objective functions is defined by $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^m$. Thus, the unconstrained robust minimization problem is defined as:

$$\min_{\mathbf{x} \in \mathbf{X}} \ \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \tag{1}$$

The worst case vector of objective function $\mathbf{f_{wc}}$ is defined as:

$$f_{\mathbf{wc}_i}(\mathbf{x}, \mathbf{P}) = \max_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \ \ i = \{1, \ldots, m\} \tag{2}$$

By using the notation in (2), solving (1) consists in finding the set of robust solutions $\mathbf{X}^*$:

$$\mathbf{X}^* = \ \ \{\mathbf{x}^* \in \mathbf{X} \mid \nexists \mathbf{x} \in \mathbf{X}, \mathbf{f_{wc}}(\mathbf{x}, \mathbf{P}) \preceq \mathbf{f_{wc}}(\mathbf{x}^*, \mathbf{P})\},$$

where the symbol $\preceq$ means $\mathbf{f_{wc}}(\mathbf{x}, \mathbf{P}) \leq \mathbf{f_{wc}}(\mathbf{x}^*, \mathbf{P})$ and $\mathbf{f_{wc}}(\mathbf{x}, \mathbf{P}) \neq \mathbf{f_{wc}}(\mathbf{x}^*, \mathbf{P})$.

The robust Pareto front $\mathbf{Y}^*$ is the image of $\mathbf{X}^*$, considering the worst case performance of the elements from $\mathbf{X}^*$, then:

$$\mathbf{Y}^* = \mathbf{f_{wc}}(\mathbf{x}^*, \mathbf{P}), \ \ \mathbf{x}^* \in \mathbf{X}^*. \tag{3}$$

The robust Pareto front $\mathbf{Y}^*$ is composed by worst case points as shown in (3). Therefore, $\mathbf{Y}^*$, or some of its elements, may be outside of the robust objective space $\mathbf{f}(\mathbf{X}, \mathbf{P})$,

$$\mathbf{f}(\mathbf{X}, \mathbf{P}) = \mathbf{f}(\mathbf{x}, \mathbf{p}), \ \ \mathbf{x} \in \mathbf{X}, \ \ \mathbf{p} \in \mathbf{P}.$$

However, this does not affect our proposed method since the worst case points are used only as reference points to

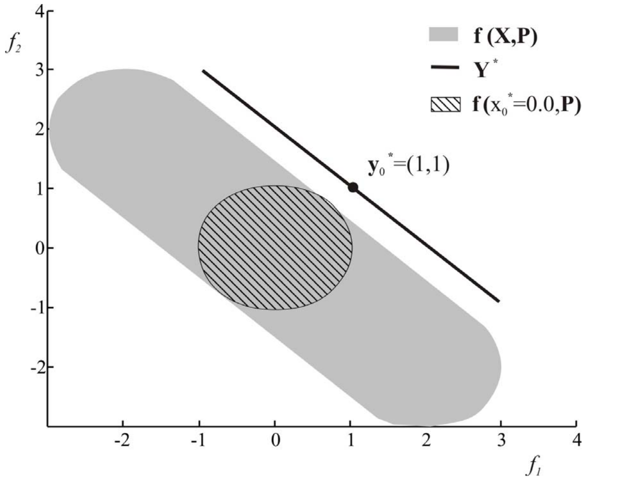

Fig. 1. The robust Pareto front to example in (4). The worst case performance of a particular solution $\mathbf{x}_0$ is highlighted. Note that $\mathbf{Y}^* \notin \mathbf{f}(\mathbf{X}, \mathbf{P})$.

guide the search to the non dominated robust solution set. The following example, illustrated in Fig. 1, shows a case where $\mathbf{Y}^* \notin \mathbf{f}(\mathbf{X}, \mathbf{P})$.

$$\min_{x \in [-2,2]} \max_{\|\mathbf{p}\| \leq 1} \quad \begin{aligned} f_1(x, p_1) &= -x + p_1, \\ f_2(x, p_2) &= x + p_2. \end{aligned}$$

Some studies treated the robust problem (1). For instance, Soares *et al.* [25] built a guaranteed wrapper over the robust Pareto front using a deterministic algorithm based only on interval methods. Also, Soares *et al.* [26] utilized an evolutionary approach to solve a constrained multi-objective electromagnetic design problem. However, in [26] the worst case design was obtained from samplings of the uncertainty parameter, not by computing the inclusion function. Rocco *et al.* [27] uses interval arithmetic concepts to guarantee the feasibility of the candidate solution in evolution strategies for mono-objective problems. However, intervals are used only to define these feasibility boxes. Regarding the present paper, it is worth remarking that we employ interval methods to compute the worst case values of the objective functions together with a MOGA to solve the robust problem (1).

## IV. Interval Multi-Objective Evolutionary Algorithm

### A. Uncertainty computation

[I]RMOEA computes the uncertainties to find the worst case performance of each objective. The process begins by bisecting the uncertainty domain $\mathbf{P}$, conveniently denoted by $[\mathbf{P}]$, in $n_p$ non-overlapping boxes $[\mathbf{p}]_i$,

$$[\mathbf{P}] = \bigcup_{i=1,\ldots,n_p} [\mathbf{p}]_i.$$

The subpaving implementation is shown in Table I. Note the stop criteria is given by the precision parameter $\varepsilon_{[\mathbf{p}]}$. Considering $\mathbf{P} \in \mathbb{R}^k$, the number of non-overlapping boxes $n_p$ is given by:

$$n_p = \prod_{i=1}^{k} \text{ceil}\Big(\frac{w([P_i])}{\varepsilon_{[\mathbf{p}]}}\Big) \tag{4}$$

where *ceil* function rounds the arguments to the nearest integers towards infinity. The non-overlapping boxes are stored in a FIFO (first in first out) queue $\mathbf{Q}_{[\mathbf{p}]}$. Thus, considering the search variable $\mathbf{x}$ and the fitness inclusion function $[\mathbf{f}](\mathbf{x}, [\mathbf{P}])$ or $[\mathbf{f}](\mathbf{x}, \mathbf{Q}_{[\mathbf{p}]})$, the minimal inclusion function $[\mathbf{f}]^*_{\mathbf{x}}$ is defined as:

TABLE I
Subpaving's generation.

```
    input ($[\mathbf{P}]$,$\varepsilon_{[\mathbf{p}]}$)
    output($\mathbf{Q}_{[\mathbf{p}]}$)

1   initialize $\mathbf{Q}_{[\mathbf{p}]}$ with $[\mathbf{P}]$
2   if $w(\mathbf{Q}_{[\mathbf{p}]}\{1\}) \leq \varepsilon_{[\mathbf{p}]}$, then go to 5
3     pull out $[\mathbf{p}]$ of $\mathbf{Q}_{[\mathbf{p}]}$
4     bisect $[\mathbf{p}]$, push in the resulting boxes in $\mathbf{Q}_{[\mathbf{p}]}$, go to 2
5   return $\mathbf{Q}_{[\mathbf{p}]}$
```

$$\overline{[f_i]}^*_{\mathbf{x}} = \max_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \overline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \ldots, n_f\}; \tag{5}$$

$$\underline{[f_i]}^*_{\mathbf{x}} = \min_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \underline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \ldots, n_f\}. \tag{6}$$

The robust image or worst case performance for $\mathbf{x}$ is the point $\overline{[f_i]}^*_{\mathbf{x}}$. Fig. 2 illustrates the worst case computation with two search variables.

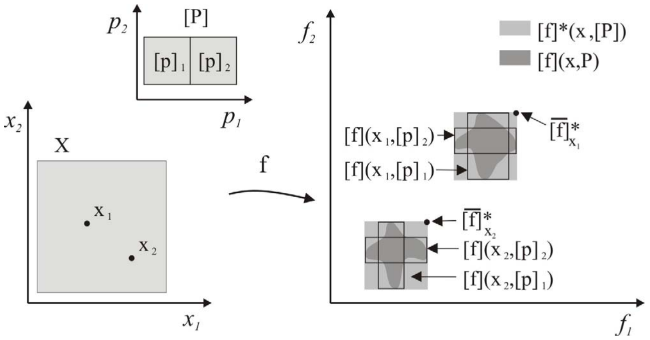

Fig. 2. The uncertainty computation and worst case examples. In dark gray, the image $\mathbf{f}(\mathbf{x}, \mathbf{P})$, obtained by point computation; in light gray, the region of $[\mathbf{f}]^*_{\mathbf{x}}$, obtained from (5) and (6). Observe the wrapping effect $\mathbf{f}(\mathbf{x}, \mathbf{P}) \subseteq [\mathbf{f}]^*_{\mathbf{x}}$. Also, note $\overline{[\mathbf{f}]}^*_{\mathbf{x}_2} \preceq \overline{[\mathbf{f}]}^*_{\mathbf{x}_1}$, i.e. $\mathbf{x}_2$ is more robust than $\mathbf{x}_1$.

**Note:** As said in the previous section, if each interval variable occurs at most once in $\mathbf{f}$, then $[\mathbf{f}]^*_{\mathbf{x}} = \mathbf{f}(\mathbf{x}, [\mathbf{P}])$, and the worst case can be obtained with only one single interval calculus. Otherwise, $[\mathbf{P}]$ has to be bisected until achieving a satisfying precision to represent $\mathbf{f}$ by intervals.

### B. [I]RMOEA

The multi-objective genetic algorithm basis to [I]RMOEA is similar to the one used by Soares *et al.* [26] except that here we treat unconstrained cases, and the uncertainty space is computed using intervals, whereas in [26] the worst case is approximated by sampling. The [I]RMOEA was implemented with the following genetic configuration: a) binary code; b) one-point crossover; c) bit-change mutation; d) roulette selection; e) a simple niche technique; and f) NSGA II elitism [1]. Options (a)-(d) are classical in genetic algorithms. In the niche technique, the objective space is divided in non-overlapping regions (niches) with the same shape and size. Each niche is stated 'valid' if it contains at least one of the $n_{pop}$ members of the population. The performance of all the $n_{niche}$ individuals in the niche $i$ are defined as $f^i_{niche} = n_{niche}$. In addition, the population is sorted into non-dominated fronts, in every generation. The performance due to the front $f^i_{front}$ is scored as 1 if the individual $i$ lies on the first front, $f^i_{front} = 2$ for the second front, and so on. Then the individual performance function $f^i_{roulette}$ is given by

$$f^i_{roulette} = \frac{1}{f^i_{niche}} + \frac{1}{f^i_{front}}.$$

The roulette selection method uses

$$\frac{f^i_{roulette}}{\sum_{j=1}^{n_{pop}} f^j_{roulette}}$$

to compute the probability of choosing individual $i$.

In step (f), the elitism strategy of the NSGA II, which is based on the union of the current population with the offspring population, is adopted. After front classification, the $n_{pop}$ first individuals are chosen to form the next parent population.

In summary, our [I]RMOEA starts with random population and the subpavings of $\mathbf{P}$, stored in $\mathbf{Q}_{[\mathbf{p}]}$. The individuals are evaluated using (5). The individuals are classified according to the front and the niche that they belong. The roulette selection is used to choose the individuals that are submitted to the crossover and mutation operations using probabilities $p_c$ and $p_m$, respectively. Then elitism strategy is applied to form the parent population of the next generation. This cycle is repeated until a convergence criteria is reached.

An important feature in the solution of robust optimization approaches is the computational effort CE. In our algorithm, it is measured in terms of objective function calls as

$$\text{CE} = (n_{gen} + 1) \times n_p \times n_{pop}, \tag{7}$$

where $n_{gen}$ is the number of generations (our convergence criteria) after the initial generation. When the fitness inclusion function has only one occurrence of each variable, then $n_p = 1$; otherwise $n_p$ is given by (4).

## V. Results

The experiments consisted in applying the [I]RMOEA to solve the robust test functions ZDT1′, ZDT2′ and ZDT3′ presented in (8), (9) and (10), respectively. These functions were adapted from the ZDT functions described in Zitzler *et al.* [21]. In the robust ZDTs, the uncertainty parameter $\mathbf{p}$ simulates the imprecision in the search parameters and in the objective functions. We used $n = 30$ variables, and the robust Pareto corresponds to $x_1 \in [0,1]$ and $x_i = 0$, $i = 2, \ldots, n$.

$$\text{ZDT1}' \begin{cases} \min\limits_{\mathbf{x}\in[\mathbf{0},\mathbf{1}]} \ \max\limits_{\mathbf{p}\in[\mathbf{0},\mathbf{0.05}]} & f_1(\mathbf{x},\mathbf{p}) = x_1 + p_1; \\ & f_2(\mathbf{x},\mathbf{p}) = h' \times h. \\ \text{where:} \quad h' = & 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h = & 1 - \sqrt{f_1/h'} + p_2. \end{cases} \tag{8}$$

$$\text{ZDT2}' \begin{cases} \min\limits_{\mathbf{x}\in[\mathbf{0},\mathbf{1}]} \ \max\limits_{p\in[-0.05,0.05]} & f_1(\mathbf{x},\mathbf{p}) = x_1; \\ & f_2(\mathbf{x},\mathbf{p}) = h' \times h. \\ \text{where:} \quad h' = & 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i + p, \\ h = & 1 - (f_1/h')^2 + p. \end{cases} \tag{9}$$

$$\text{ZDT3}' \begin{cases} \min\limits_{\mathbf{x}\in[\mathbf{0},\mathbf{1}]} \ \max\limits_{p\in[-0.1,0.1]} & f_1(\mathbf{x},\mathbf{p}) = x_1; \\ & f_2(\mathbf{x},\mathbf{p}) = h' \times h. \\ \text{where:} \quad h' = & 1 + \frac{9}{n-1}(\sum_{i=2}^{n} x_i + p), \\ h = & 1 - \sqrt{\frac{f_1}{h'}} - \frac{f_1}{h'}\sin(10\pi f_1). \end{cases} \tag{10}$$

In our experiments, we have used: $n_{pop} = 50, 100, \ldots, 250$ individuals; $n_{gen} = 100, 150, \ldots, 300$ generations; code with 20 bits per variable; $p_m = 0.001$; and $p_c = 1$. Each parameter configuration was executed only one time. Also, $\varepsilon_{[\mathbf{p}]} = 0.01, 0.025, 0.05$ to ZDT1′, $\varepsilon_{[\mathbf{p}]} = 0.01, 0.025, 0.05, 0.1$ to ZDT2′, and $\varepsilon_{[\mathbf{p}]} = 0.025, 0.05, 0.1, 0.2$ to ZDT3′. Some interesting results are presented in Figs. 3 to 5. These figures show three curves: a) the robust Pareto front obtained by [I]RMOEA; b) the true robust Pareto front; and c) the original (non robust) Pareto front, such as in Zitzler *et al.* [21].

Fig. 3, 4 and 5 illustrate the effect of the uncertainty parameter in the robust ZDT functions. Since we considered minimization in worst case approaches, the robust Pareto fronts must be located above the non robust ones, as showed in the figures. Interesting to note that the images of the solution set, provided by [I]RMOEA, are above the robust Pareto front. This must be true because interval methods are always pessimistic, see again the wrapping effect in Fig. 2.

In the robust ZDTs, $\mathbf{p}$ may have any distribution form, that is, it is not important to [I]RMOEA if $\mathbf{p}$ is Gaussian or not because the complete uncertainty space was computed using interval mechanisms. Although the wrapping effect can be too pessimistic for the results, the subpavings (see Table I) may be created to reduce the pessimism. Unfortunately, the subpavings give an additional overhead to [I]RMOEA since the computation effort, given by (7), increase $n_p$ times. As one can note, the $n_p$ plays the same role of the number of samples in probabilistic methods. In each ZDT experiment,

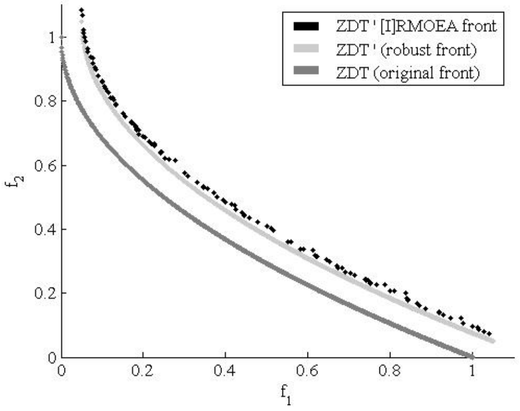

Fig. 3. Pareto fronts of ZDT1 and ZDT1$'$. Simulation details: $n_{pop} = 200$; $n_{gen} = 250$; $\varepsilon_{[\mathbf{p}]} = 0.05$; and CE=50200 function calls.

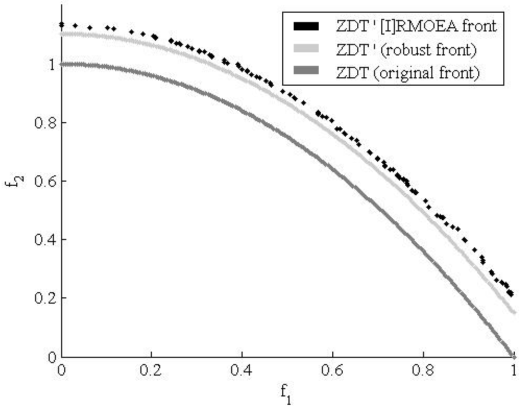

Fig. 4. Pareto fronts of ZDT2 and ZDT2$'$. Simulation details: $n_{pop} = 200$; $n_{gen} = 250$; $\varepsilon_{[\mathbf{p}]} = 0.1$; and CE=50200 function calls.

we have used some different values to $n_p$ obtained using $\varepsilon_{[\mathbf{p}]}$ sets described before in (4). We observed similar results in the graphics even in those with higher $n_p$, showing that the subpaving computation is not necessary in these cases. Some reasons are given in the following. First, because $\mathbf{p}$ occurs only once in ZDT1$'$ and the theorem in [7] (p.30) states that the inclusion function has minimal width, thus obtained with only one calculus using the whole interval. Second, the computation of $\mathbf{p}$ in ZDT2$'$ and in ZDT3$'$ did not cause pessimistic results because of the additive way the $[\mathbf{p}]$ is inserted in the functions. Considering that we obtained the robust results without using subpavings ($n_p$=1), we can say that [I]RMOEA achieved minimal computational effort.

## VI. Conclusions

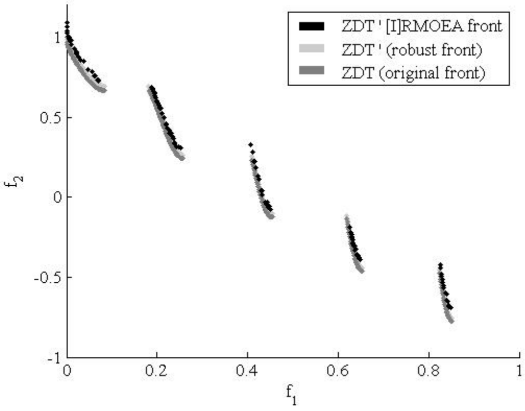

Fig. 5. Pareto fronts of ZDT3 and ZDT3$'$. Simulation details: $n_{pop} = 200$; $n_{gen} = 250$; $\varepsilon_{[\mathbf{p}]} = 0.2$; and CE=50200 function calls.

Uncertainties are always present in optimization problems and when their effects are significant, it is necessary to handle them in the design stage. Some ways to interpret and treat uncertainties have been discussed, and we have chosen the worst case formulation in our proposed algorithm. [I]RMOEA uses the power of the evolutionary search engine to execute the minimization process and the techniques of interval analysis to compute the worst case performance. With intervals, at least two interesting advantages can be mentioned: a) the effects of uncertainties can be completely considered, such that robustness is a question of precision not of approximation; and b) the worst case performance can be achieved using a single interval, when the fitness inclusion function has no dependent factors, such as multiple occurrences of interval parameters. The [I]RMOEA was tested in the ZDT robust test functions with 30 variables. The results show that these cases are computationally tractable, and hence we find [I]RMOEA a promising technique to be extended to another robust formulations.

## Acknowledgement

This work was supported by CNPq (Grant 307990/2006-0) and PUCMinas from Brazil.

## References

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[2] J. D. Knowles, "Local search and hybrid evolutionary algorithms for pareto optimization," Ph.D. dissertation, Departament of Computer Science, University of Reading, Reading, UK, 2002.

[3] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study and the strength pareto approach," in *Parallel Problem Solving from Nature V (PPSN-V)*, 1998, pp. 292–301.

[4] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *Evolutionary Computation, IEEE Transactions on*, vol. 9, pp. 303– 317, 2005.

[5] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[6] R. Moore, *Dimensional Analysis*, E. Cliffs, Ed. New Jersey: Prentice Hall, 1966.

[7] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.

[8] E. Hansen and G. Walster, *Global Optimization Using Interval Analysis*, 2nd ed. Marcel Dekker, Inc, 2004.

[9] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 405–420, 2006.

[10] Y. Jin and B. Sendhoff, “Trade-off between performance and robustness: An evolutionary multiobjective approach,” in *Proceedings of the Second International Conference on Evolutionary Multicriterion Optimization, LNCS*, vol. 2632. Springer, 2003, pp. 237–251.

[11] F. G. Guimaraes, D. A. Lowther, and J. A. Ramirez, “Multiobjective approaches for robust electromagnetic design,” *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1207–1210, 2006.

[12] K. Deb and H. Gupta, “Searching for robust pareto-optimal solutions in multi-objective optimization.” in *EMO*, 2005, pp. 150–164.

[13] L. Bui, H. Abbass, and D. Essam, “Fitness inheritance for noisy evolutionary multi-objective optimization,” in *Genetic and Evolutionary Computation Conference (GECCO)*. Washington D.C.: ACM Press, 2005.

[14] H. Kitano, “Biological robustness,” *Nature Reviews - Genetics*, vol. 5, pp. 826–837, 2004.

[15] J. DeVale and P. J. Koopman Jr., “Robust software - no more excuses,” in *DSN ’02: Proceedings of the 2002 International Conference on Dependable Systems and Networks*. IEEE Computer Society, 2002, pp. 145–154.

[16] X. Du and W. Chen, “Towards a better understanding of modeling feasibility robustness in engineering design,” 1999.

[17] A. Parkinson, C. Sorensen, and N. Pourhassan, “A general approach for robust optimal design." journal of mechanical design,” *Transactions of the ASME*, vol. 115, no. 1, pp. 75–81, 1993.

[18] P. Kouvelis and G. Yu, *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.

[19] B. Roy, “A missing link in OR-DA, robustness analysis,” *Foundations of Computing and Decision Sciences*, vol. 23, no. 3, pp. 141–160, 1998.

[20] L. C. Dias and J. N. Clímaco, “On computing electre’s credibility indices under partial information,” *Journal of Multi-Criteria Decision Analysis*, vol. 8, pp. 74–92, 1999.

[21] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evolutionary Computation Journal*, vol. 8, no. 2, pp. 125–148, 2000.

[22] C. A. C. Coello, “Evolutionary multi-objective optimization: a historical view of the field,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[23] H. Ratschek and J. Rokne, *Computer Methods for the Range of Functions*. Halsted Press, 1984.

[24] G. Chabert and L. Jaulin, “Computing the pessimism of inclusion functions,” *Reliable computing*, vol. 13, no. 6, pp. 489–504, 2007.

[25] G. L. Soares, R. O. Parreiras, L. Jaulin, C. A. Maia, and J. A. Vasconcelos, “Interval robust multi-objective algorithm,” to appear in Nonlinear Analysis: Theory, Methods & Applications.

[26] G. L. Soares, R. L. Adriano, C. A. Maia, L. Jaulin, and J. A. Vasconcelos, “Robust multi-objective team 22 problem: a case study of uncertainties in design optimization,” to appear in IEEE Transactions on Magnetics.

[27] C. M. Rocco, J. A. Moreno, and N. Carrasquero, “Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application,” *Reliability Engineering and System Safety*, vol. 79, no. 2, pp. 149–159, 2003.