1. Multiobjective problem

The MO problem can be formulated as follows:

$$\min \mathbf{J}(\theta) = \min[J_1(\theta), J_2(\theta), \ldots, J_s(\theta)] \tag{1}$$

subject to:

$$\begin{aligned} g_q(\theta) \leq 0, & \qquad (1 \leq q \leq r) \\ h_k(\theta) = 0, & \qquad (1 \leq k \leq n) \\ \theta_{li} \leq \theta_i \leq \theta_{ui}, & \qquad (1 \leq i \leq L) \end{aligned} \tag{2}$$

where $J_i(\theta)$, $i \in B := [1 \ldots s]$ are the objectives to be optimized, θ is a solution inside the L-dimensional solution space D, $g_q(\theta)$ and $h_k(\theta)$ are each of the r inequality and n equality problem constraints respectively and θ_{li} and θ_{ui} are the lower and upper constraints which defined the solution space D.

To solve the MO problem the Pareto optimal set $\mathbf{\Theta}_P$ (solutions where none of them dominate any of the others) must be found. Pareto dominance is defined as follows.

A solution θ^1 dominates another solution θ^2, denoted by $\theta_1 \prec \theta_2$, iff

$$\forall i \in B, J_i(\theta^1) \leq J_i(\theta^2) \ \wedge \ \exists k \in B : J_k(\theta^1) < J_k(\theta^2) \ .$$

Therefore the Pareto optimal set $\mathbf{\Theta}_P$ is given by

$$\mathbf{\Theta}_P = \{\theta \in D \mid \nexists \ \ \tilde{\theta} \in D \ : \ \tilde{\theta} \prec \theta\} \ . \tag{3}$$

$\mathbf{\Theta}_P$ is unique and normally includes infinite solutions. Hence a set $\mathbf{\Theta}_P^*$, with a finite number of elements from $\mathbf{\Theta}_P$, should be obtained[1].

[1]Notice that $\mathbf{\Theta}_P^*$ is not unique.

2. ϵ-MOGA

The ϵ-MOGA variable (ev-MOGA)[1] is an elitist multi-objective evolutionary algorithm [2] based on the concept of ϵ-dominance [3], which is used to control the content of the archive $A(t)$ where the result of the optimization problem is stored. ev-MOGA tries to ensure that $A(t)$ converges toward an ϵ-Pareto set, $\mathbf{\Theta}^*_{P\epsilon}$ in a smart distributed manner along the Pareto front $\mathbf{J}(\mathbf{\Theta}_P)$ with limited memory resources. It also adjusts the limits of the Pareto front dynamically and prevents the solutions belonging to the ends of the front from being lost.

For this reason, the objective space is split into a fixed number of boxes. For each dimension $i \in B$, n_box_i cells of ϵ_i width are created where

$$\epsilon_i = (J_i^{max} - J_i^{min})/n\_box_i,$$

$$J_i^{max} = \max_{\theta \in \mathbf{\Theta}^*_{P\epsilon}} J_i(\theta), \; J_i^{min} = \min_{\theta \in \mathbf{\Theta}^*_{P\epsilon}} J_i(\theta).$$

This grid preserves the diversity of $\mathbf{J}(\mathbf{\Theta}^*_{P\epsilon})$ since each box can be occupied by only one solution in $A(t)$ and at the same time produces a smart distribution as will be shown later[2].

The concept of ϵ-dominance is defined as follows. For a solution $\theta \in D$, $box_i(\theta)$ is defined by

$$box_i(\theta) = \left\lceil \frac{J_i(\theta) - J_i^{min}}{J_i^{max} - J_i^{min}} \cdot n\_box_i \right\rceil \; \forall i \in B. \tag{4}$$

[2]The algorithm only checks occupied boxes (not all boxes). This content management of $A(t)$ avoids the needs to use other clustering techniques to obtain adequate distributions, and so considerably reduces the computational cost (see reference [3]).

Let $\mathbf{box}(\theta) = \{box_1(\theta), \ldots, box_s(\theta)\}$. A solution θ^1 with value $\mathbf{J}(\theta^1)$ ϵ-dominates the solution θ^2 with value $J(\theta^2)$, denoted by $\theta^1 \prec_\epsilon \theta^2$, if and only if

$$\mathbf{box}(\theta^1) \prec \mathbf{box}(\theta^2) \vee \left(\mathbf{box}(\theta^1) = \mathbf{box}(\theta^2) and \theta^1 \prec \theta^2\right) .$$

Hence, a set $\mathbf{\Theta}^*_{P\epsilon} \subseteq \mathbf{\Theta}_P$ is ϵ-Pareto if and only if

$$\forall \theta^1, \theta^2 \in \mathbf{\Theta}^*_{P\epsilon}, \theta^1 \neq \theta^2,\ \mathbf{box}(\theta^1) \neq \mathbf{box}(\theta^2) and \mathbf{box}(\theta^1) \not\prec_\epsilon \mathbf{box}(\theta^2) \qquad (5)$$

Therefore, ev-MOGA is responsible for updating the content of $A(t)$ by saving only ϵ-dominant solutions that do not share the same box. When two mutually ϵ-dominant solutions compete, the solution that prevails in $A(t)$ will be the one that is closest to the center of the box. It is thereby possible to prevent solutions belonging to adjacent boxes (neither of them dominating the other) from being too close to each other, thus encouraging a smart distribution.

The aim of ev-MOGA is to achieve a $\mathbf{\Theta}^*_{P\epsilon}$ with the greatest possible number of solutions in order to characterize the Pareto front adequately. Although the number of possible solutions will depend on the shape of the front and on n_box_i, it will not exceed the following level

$$|\Theta^*_{P_\epsilon}| \leq \frac{\prod_{i=1}^{n} n\_box_i + 1}{n\_box_{max} + 1}, \quad n\_box_{max} = \max_i n\_box_i \qquad (6)$$

which is advantageous, as it is possible to control the maximum number of solutions that will characterize the Pareto front.

Furthermore, thanks to the definition of box, the anchor points $J_i(\theta^{i*})$ are assigned a value of $box_i(\theta^{i*}) = 0$, whereby $J_i(\theta^{i*}) = J_i^{min}$. Therefore, no solution θ can ϵ-dominate them because, by applying the definition of box, their $box_i(\theta) \geq 1$.

Figure 1 shows what $\mathbf{\Theta}^*_{P\epsilon}$ would be if obtained by applying concepts of ϵ-dominance for a bi-objective example, when $n\_box_1 = n\_box_2 = 10$ is used. The values ϵ_1 and ϵ_2 depend on the limits of the front J_1^{min}, J_2^{min}, J_1^{max} and J_2^{max}, which adjust dynamically in accordance with the solutions as they are located. It can be seen that the distribution of solutions comprised by $J(\mathbf{\Theta}^*_{P\epsilon})$, along the front, depends on its slope, the greatest number of points accumulating in the central area (indicated by a dotted line) where the slope is greatest.

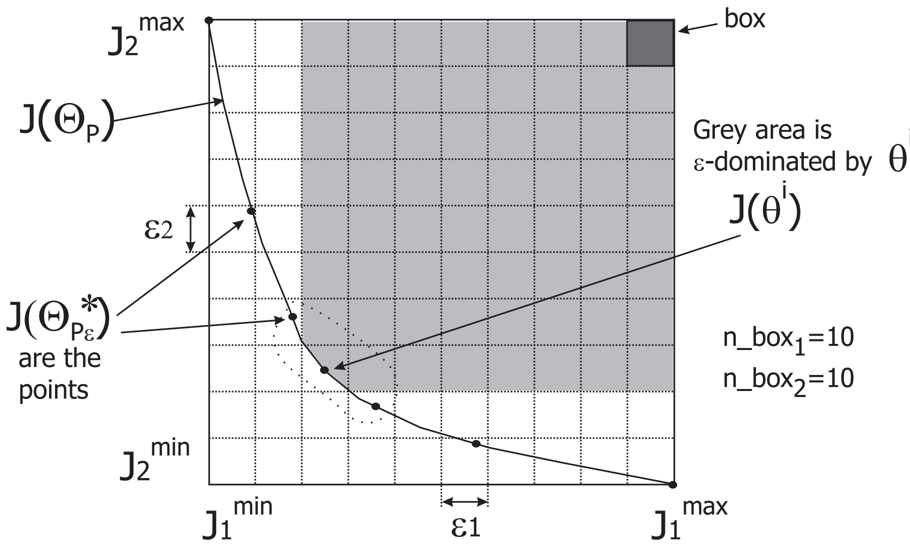

Figure 1: The concept of ϵ-dominance. ϵ-Pareto front $J(\mathbf{\Theta}^*_{P\epsilon})$ in a bi-objective problem. J_1^{min}, J_2^{min}, J_1^{max}, J_2^{max}, Pareto front limits; ϵ_1, ϵ_2 box widths; and n_box_1, n_box_2, number of boxes for each dimension.

A description of the ev-MOGA algorithm for obtaining an ϵ-Pareto front $\mathbf{J}(\mathbf{\Theta}^*_{P\epsilon})$, is presented below. The algorithm, which adjusts the width ϵ_i dynamically, is composed of three populations:

1. Main population $P(t)$ explores the searching space D during the algorithm iterations (t). Population size is $Nind_P$.
2. Archive $A(t)$ stores the solution $\boldsymbol{\Theta}^*_{P\epsilon}$. Its size $Nind_A$ is variable but bounded (see equation (6)).
3. Auxiliary population $G(t)$. Its size is $Nind_G$, which must be an even number.

The pseudocode of the ev-MOGA algorithm is given by

```
1.  t:=0
2.  A(t):=∅
3.  P(t):=ini_random(D)
4.  eval(P(t))
5.  A(t):=store_ini(P(t),A(t))
6.  while t<t_max do
7.      G(t):=create(P(t),A(t))
8.      eval(G(t))
9.      A(t+1):=store(G(t),A(t))
10.     P(t+1):=update(G(t),P(t))
11.     t:=t+1
12. end while
```

The main steps of the algorithm are detailed as follows:

Step 3. $P(0)$ is initialized with $Nind_P$ individuals (solutions) that have been randomly selected from the searching space D.

Step 4 and 8. Function **eval** calculates the function value (Equation (1)) for each individual in $P(t)$ (step 4) and $G(t)$ (step 8).

Step 5. Function $\mathbf{store}_{ini}$ checks individuals in $P(t)$ that might be included in the archive $A(t)$ as follows:

1. Non-dominated $P(t)$ individuals are detected, $\mathbf{\Theta}_{ND}$.
2. Pareto front limits J_i^{max} and J_i^{min} are calculated from $\mathbf{J}(\theta), \forall\theta \in \mathbf{\Theta}_{ND}$.
3. Individuals in $\mathbf{\Theta}_{ND}$ are analyzed, one by one, and those that are not ϵ-dominated by individuals in $A(t)$, will be included in $A(t)$.

Step 7. With each iteration, the function **create** creates $G(t)$ as follows:

1. Two individuals are randomly selected, θ^P from $P(t)$ and θ^A from $A(t)$.
2. A random number $u \in [0 \ldots 1]$ is generated.
3. If $u > P_{c/m}$ (probability of crossing/mutation), θ^P and θ^A are crossed over by means of the extended linear recombination technique [4].
4. If $u \leq P_{c/m}$, θ^P and θ^A are mutated using random mutation with Gaussian distribution [4] and then included in $G(t)$.

This procedure is repeated $Nind_G/2$ times until $G(t)$ is filled.

Step 9. Function **store** checks, one by one, which individuals in $G(t)$ must be included in $A(t)$ on the basis of their location in the objective space (see Figure 2). Thus $\forall\, \theta^G \in G(t)$

1. If $J(\theta^G)$ belongs to the area $Z1$ and is not ϵ-dominated by any individual from $A(t)$, it will be included in $A(t)$ (if its box is occupied by an individual that is also not ϵ-dominated, then the individual lying furthest away from the center box will be eliminated). Individuals from $A(t)$ which are ϵ-dominated by θ^G will be eliminated.
2. If $J(\theta^G)$ belongs to the area $Z2$ then it is not included in the archive, since it is dominated by all individuals in $A(t)$.
3. If $J(\theta^G)$ belongs to the area $Z3$, the same procedure is applied as was used with the function $\mathbf{store}_{ini}$ but now applied over a population $P'(t) = A(t) \bigcup \theta^G$, that is, $\mathbf{store}_{ini}(P'(t), \emptyset)$. In this procedure new Pareto front limits and ϵ_i widths could be recalculated.
4. If $J(\theta^G)$ belongs to the area $Z4$, all individuals from $A(t)$ are deleted since they are all ϵ-dominated by θ^G. θ^G is included and the objective space limits are $\mathbf{J}(\theta^G)$.

Step 10. Function **update** updates $P(t)$ with individuals from $G(t)$. Every individual θ^G from $G(t)$ is compared with an individual θ^P that is randomly selected from the individuals in $P(t)$ which are dominated by θ^G. θ^G will not be included in $P(t)$ if thre is no individual in $P(t)$ dominated by θ^G.

Finally, individuals from $A(t)$ comprise $\mathbf{\Theta}^*_{P\epsilon}$, the smart characterization of the Pareto front.

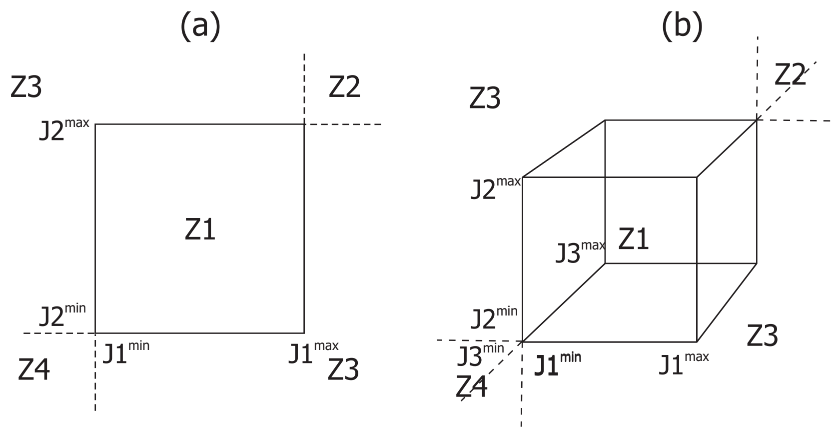

Figure 2: Function space areas (Z) and limits (J). (a) two-dimensional case; (b) tri-dimensional case.

[1] Herrero JM. Non-linear Robust identification using evolutionary algorithms, Ph.D. Thesis, Polytechnic University of Valencia, 2006.

[2] Coello C, Toscano G, Mezura E. Current and future research trends in evolutionary multi-objective optimization. In: Manuel Graa, Richard Duro, Alicia dnjou, and Paul P. Wang, editors. Information Processing with Evolutionary Algorithms: From Industrial Applications to Academic Speculations. Springer-Verlag 2005; 213-231.

[3] Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multi-objective optimization. Evolutionary computation 2002; 10(3):263-282.

[4] Herrero JM, Blasco X, Martnez M, Sanchis J: Robust identification of a biomedical process by evolutionary algorithms (in Spanish). Revista Iberoamericana de Automtica e Informtica Industria 2006; 3(4):75-86