

Document downloaded from:

<http://hdl.handle.net/10251/57265>

This paper must be cited as:

Reynoso Meza, G.; Sanchís Saez, J.; Blasco Ferragud, FX.; Garcia Nieto, S. (2014). Physical programming for preference driven evolutionary multi-objective optimization. *Applied Soft Computing*. 24:341-362. doi:10.1016/j.asoc.2014.07.009.



The final publication is available at

Copyright Elsevier

Additional Information

# Physical Programming for Preference Driven Evolutionary Multi-Objective Optimisation

Gilberto Reynoso-Meza <sup>a,\*</sup>, Javier Sanchis <sup>a</sup>, Xavier Blasco <sup>a</sup>, Sergio García-Nieto <sup>a</sup>

<sup>a</sup>*Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera s/n , Valencia 46022, Spain*

---

## Abstract

Preference articulation in multi-objective optimisation could be used to improve the pertinency of solutions in an approximated Pareto front. That is, computing the most interesting solutions from the designer's point of view in order to facilitate the Pareto front analysis and the selection of a design alternative. This articulation can be achieved in an *a priori*, *progressive*, or *a posteriori* manner. If it is used within an *a priori* frame, it could focus the optimisation process towards the most promising areas of the Pareto front, saving computational resources and assuring a useful Pareto front approximation for the designer. In this work, a physical programming approach embedded in an evolutionary multi-objective optimisation is presented as a tool for preference inclusion. The results presented and the algorithm developed validate the proposal as a potential tool for engineering design by means of evolutionary multi-objective optimisation.

*Keywords:* Multi-objective optimisation design procedure, evolutionary multi-objective optimisation, physical programming, many-objective optimisation, preference articulation, decision making.

---

## 1. Introduction

Multi-objective optimisation design (MOOD) procedures are *generate first choose later* (GFCL) holistic strategies for multi-objective problems [1]. A multi-objective problem (MOP) arises when multiple objectives and requirements must be fulfilled by the designer. Such objectives are usually in conflict with each other; therefore a trade-off solution must be calculated and selected for implementation. The GFCL strategy generates a set of potentially preferable design alternatives and the decision maker (DM or simply the designer) selects the most preferable solution according to his or her preferences. These solutions are generally Pareto optimal solutions [2].

The MOOD procedure (see Figure 1) identifies three main (possibly fundamental) steps [3, 4]: the MOP definition (measure); the multi-objective optimisation process (search); and

---

\*Corresponding author. Tel.:+34963877007;fax:+34963879579  
Email address: [gilreyme@upv.es](mailto:gilreyme@upv.es) (Gilberto Reynoso-Meza )  
URL: <http://cpoh.upv.es/> (Gilberto Reynoso-Meza )

the multi-criteria decision making (MCDM) step (decision making). Major efforts are made to improve the algorithms and tools to facilitate the two latter processes.

In the case of multi-objective optimisation, several algorithms have been designed (NBI [5], NNC<sup>1</sup> [6, 7], NSGA-II<sup>2</sup> [8], MOGA<sup>3</sup> [9], MOEA/D<sup>4</sup> [10] for example) and used in a wide variety of applications [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Such algorithms mainly seek a set of Pareto optimal solutions that describe a Pareto front approximation. According to the designer's wishes, those algorithms would incorporate some desirable characteristics [23] such as convergence (capacity to reach the Pareto front), diversity (capacity to generate different solutions), and pertinency (capacity to generate useful solutions for the DM). For the decision making step, several tools and visualisations approaches [25] have been proposed over the years (scatter plot diagrams, parallel coordinates [26], level diagrams [27, 28] or self-organizing maps [29] for example).

In [30, 31] the importance of considering both processes (optimisation and selection) in a holistic way, in order to guarantee a full embedment of the DM in the decision making step, was noted. This is because the decision making process is usually more time consuming than the optimisation step [32]. This embedment could be achieved by providing a useful set of solutions to the designer; thereby analysing the trade-off between conflicting objectives in order to refine his or her final selection [30].

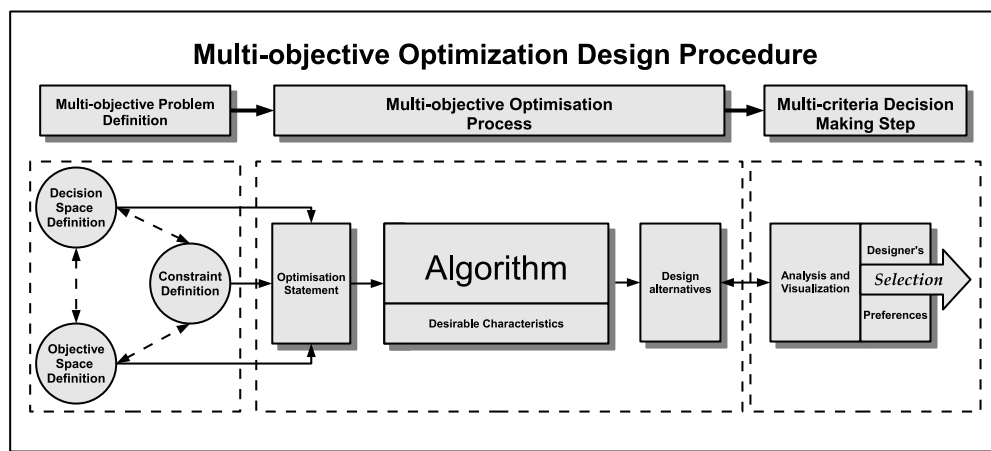


Figure 1: A multi-objective optimisation design (MOOD) procedure.

Given that the MOOD procedure should be a holistic technique, preference handling mechanisms could play a major role in bridging the gap between optimisation and the selection process. These mechanisms will enable the algorithm to approximate a Pareto front with pertinent solutions in the search process; and therefore facilitating the DM's task of

<sup>1</sup>Matlab code available at <http://www.mathworks.com/matlabcentral/fileexchange/38976>

<sup>2</sup>Source code available at: <http://www.iitk.ac.in/kangal/codes.shtml>; also, a variant of this algorithm is available in the global optimisation toolbox of Matlab.

<sup>3</sup>Toolbox for Matlab available at <http://www.sheffield.ac.uk/acse/research/ecrg/gat>

<sup>4</sup>Matlab code available at <http://cswww.essex.ac.uk/staff/zhang/IntrotoResearch/MOEAd.htm>

analysing and selecting a design alternative [33]. Furthermore, preference handling might be used in constrained optimisation instances and many-objective optimisation statements [34]. Challenges for preference articulation include building a practical framework to link the designer’s desired trade-off with the cost function to optimise.

A first step for the aforementioned challenge, is stating meaningful design objectives. Sometimes with classical optimisation approaches a cost function (or objective) is built in order to satisfy a set of requirements such as convexity and/or continuity; that is, it is built from the point of view of the optimiser despite a possible loss of interpretability. The usage of more interpretable objectives facilitates the inclusion of preferences in the optimisation process, producing meaningful and pertinent solutions for the designer in the selection step. Evolutionary multi-objective optimisation (EMO) provides a helpful framework for this purpose, since multi-objective evolutionary algorithms (MOEAs) have been shown to be a flexible tool to handle constrained complex functions [32] in a wide variety of engineering domain applications [11]. Furthermore, a convenient feature of using MOEAs is the possibility of selecting more interpretable objectives for the designer. That is, the objective selection could be closer to the point of view of the designer, rather than the optimiser. Nevertheless, this is just a necessary step to moving forward to preference articulation, since this could assure meaningful, but not pertinent, design alternatives.

The physical programming (PP) method [35] is very suitable for multi-objective engineering design since it formulates design objectives in an understandable and intuitive language for designers. Since it defines desirable, tolerable, and undesirable ranges for individual objectives, it becomes a potential technique to improve the pertinency of solutions in multi-objective optimisation. PP has been merged previously with classical optimisation techniques [1, 36]; nevertheless, it remains an interesting topic to merge with MOEAs.

In this work, PP is merged with MOEAs as an auxiliary mechanism to improve the pertinency of the calculated solutions. Such an approach will enable the DM to have more useful solutions, since it provides a flexible and intuitive coding framework where the MOP is built from the DM’s point of view. Although an algorithm to test its viability is developed, it could be incorporated in other MOEAs. The remainder of this work is as follows: in Section 2 several preliminaries in multi-objective optimisation, physical programming, and the MOEA are presented. In Section 3 the preference handling mechanism is explained, and then evaluated in Section 4. Finally, some concluding remarks are given.

## 2. Background

To state the proposal, some notions on multi-objective optimisation, preference handling, physical programming, and the algorithm to be used are required. Those are provided below.

### 2.1. Multi-objective optimisation review

As referred in [2], a MOP <sup>5</sup>, can be stated as follows:

---

<sup>5</sup>A maximization problem can be converted to a minimization problem. For each of the objectives that have to be maximized, the transformation:  $\max J_i(\boldsymbol{\theta}) = -\min(-J_i(\boldsymbol{\theta}))$  could be applied.

$$\min_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \dots, J_m(\boldsymbol{\theta})] \quad (1)$$

subject to:

$$\mathbf{K}(\boldsymbol{\theta}) \leq 0 \quad (2)$$

$$\mathbf{L}(\boldsymbol{\theta}) = 0 \quad (3)$$

$$\underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, i = [1, \dots, n] \quad (4)$$

where  $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$  is defined as the decision vector with  $\dim(\boldsymbol{\theta}) = n$ ;  $\mathbf{J}(\boldsymbol{\theta})$  as the objective vector and  $\mathbf{K}(\boldsymbol{\theta})$ ,  $\mathbf{L}(\boldsymbol{\theta})$  as the inequality and equality constraint vectors respectively;  $\underline{\theta}_i, \bar{\theta}_i$  are the lower and upper bounds in the decision space.

It has been pointed out that there is not a single solution in MOPs, because there is not generally a better solution in all the objectives. Therefore, a set of solutions, the Pareto set, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front. All the solutions in the Pareto front are a set of Pareto optimal and non-dominated solutions:

**Definition 1.** (*Pareto optimality [2]*): An objective vector  $\mathbf{J}(\boldsymbol{\theta}^1)$  is Pareto optimal if there does not exist another objective vector  $\mathbf{J}(\boldsymbol{\theta}^2)$  such that  $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$  for all  $i \in [1, 2, \dots, m]$  and  $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$  for at least one  $j$ ,  $j \in [1, 2, \dots, m]$ .

**Definition 2.** (*Dominance [11]*): An objective vector  $\mathbf{J}(\boldsymbol{\theta}^1)$  is dominated by another objective vector  $\mathbf{J}(\boldsymbol{\theta}^2)$  iff  $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$  for all  $i \in [1, 2, \dots, m]$  and  $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$  for at least one  $j$ ,  $j \in [1, 2, \dots, m]$ . This is denoted as  $\mathbf{J}(\boldsymbol{\theta}^2) \preceq \mathbf{J}(\boldsymbol{\theta}^1)$ .

**Definition 3.** (*Strict dominance*): An objective vector  $\mathbf{J}(\boldsymbol{\theta}^1)$  is strictly dominated by another objective vector  $\mathbf{J}(\boldsymbol{\theta}^2)$  if  $J_i(\boldsymbol{\theta}^2) < J_i(\boldsymbol{\theta}^1)$  for all  $i \in [1, 2, \dots, m]$ . This is denoted as  $\mathbf{J}(\boldsymbol{\theta}^2) \prec \mathbf{J}(\boldsymbol{\theta}^1)$ .

For example, in Figure 2, five different solutions ( $\diamond$ ) are calculated in order to approximate a Pareto front (bold line). Solutions A, B and C are non-dominated solutions, since there is not a better solution vector (in the calculated set) for all the objectives. Solutions B and C are not Pareto optimal, since some solutions (not found in this case) (strictly) dominate them. Furthermore, solution A is also Pareto optimal, since it lies on the feasible Pareto front. The set of non-dominated solutions built the Pareto front approximation  $\mathbf{J}_P^*$ . It is important to notice that the Pareto front is usually unknown, and the DM can only rely on Pareto front approximations.

## 2.2. Background on preference handling in multi-objective optimisation

As commented before, one potentially desirable characteristic of a MOEA is the mechanism for preference handling in order to calculate pertinent solutions. That is, the capacity to obtain a set of interesting solutions from the DM's point of view (Figure 3). Incorporating

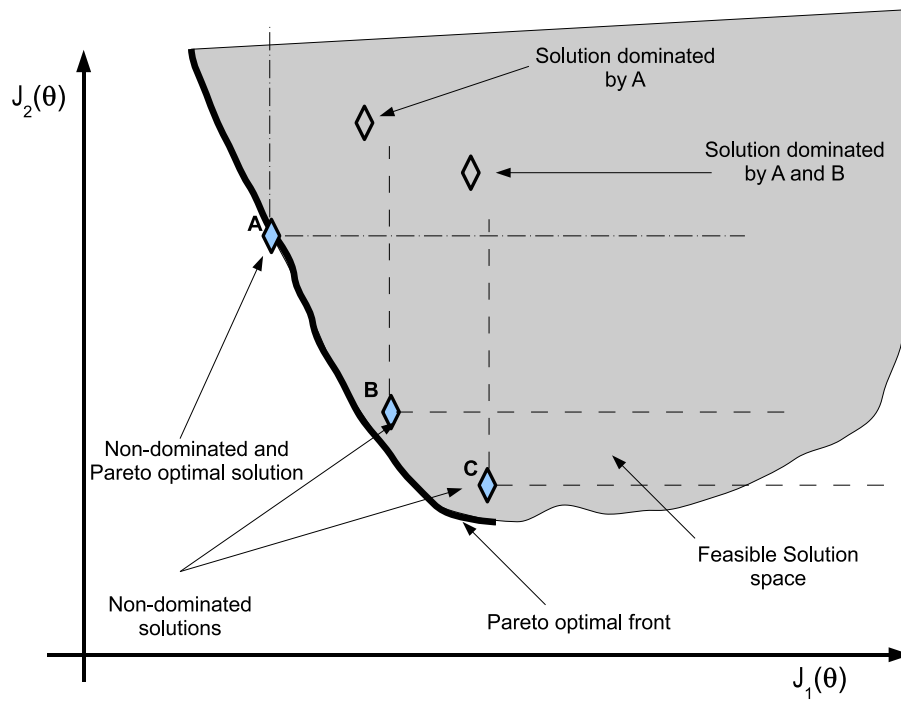


Figure 2: Pareto optimality and dominance concepts.

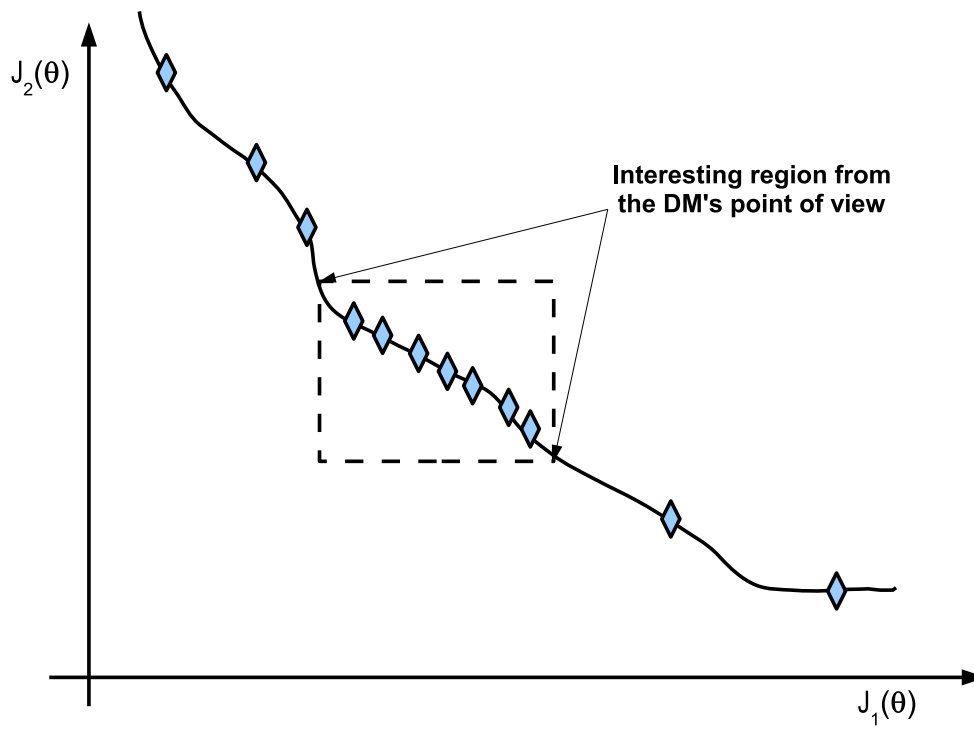


Figure 3: Pertinency concept of design alternatives.

the DM's preferences into MOEAs has been suggested to improve the pertinency of solutions (see for example [33, 37]).

The designer's preferences could be defined in the MOOD procedure in an *a priori*, *progressive*, or *a posteriori* fashion [38].

- *A priori*: the designer has partial knowledge about his or her preferences in the design objective space. In such cases, the DM could be interested in using an algorithm that enables incorporating such preferences in the optimisation procedure.
- *Progressive*: the optimisation algorithm embeds the designer into the optimisation process to adjust or change his or her preferences *on the fly*. This could be a desirable characteristic for an algorithm when the designer has some knowledge of the objective trade-off in complex problems.
- *A posteriori*: the designer analyses the Pareto front calculated by the algorithm, and according to the set of solutions, he or she defines the preferences in order to select a preferable solution.

It is also possible to classify preference handling techniques into five classes with respect to the question: *what it is important for the designer?*:

***Dominance is essential***: it is important for the designer to calculate a set of solutions that dominate one or more *reference* objective vectors.

***Objective against objective***: it is important for the designer to identify which objectives have priority over others through the EMO process.

***Objective value against objective value***: it is important for the designer to identify when the value of a given objective has priority over the value of others.

***Subset against subset***: identifying a combination of objectives and values that are preferred over others.

Some popular techniques include ranking procedures, goal attainment, and fuzzy relations [33]. Possibly one of the first algorithms to include preference information is the MOGA [39, 40] algorithm which uses a goal vector scheme. In [41], the NSGA-II algorithm is improved using preferences in a fuzzy scheme. Other examples are presented in [42] where a preference information approach was merged with the IBEA proposal; or in [40] where a preference articulation technique is used in the MOGA framework. Examples where the ranking scheme has been used include [43, 44, 45, 46]. In any case, some desirable characteristics of the preference handling mechanism have been stated in [47]:

- Handling multiple preference conditions simultaneously.
- Approximating Pareto-optimal solutions for each preference condition.

- The handling mechanism should be indifferent to the shape of the Pareto front.
- It should be capable of handling many-objective optimisation instances.

Here, the following feature is also considered:

- It should enable the DM to decide how many solutions are required in the Pareto front approximation, which will be analysed in the MCDM step.

In this paper, an implementation using physical programming as an *a priori* technique for preference handling in EMO is presented. Such implementation will incorporate modifications in order to cover the features discussed above. A brief review of physical programming is presented below.

### 2.3. Physical programming review

The physical programming (PP) method is a suitable technique for multi-objective engineering design since it formulates design objectives in an understandable and intuitive language for designers. PP is an aggregate objective function (AOF) technique [31] for multi-objective problems that includes the available information in the optimisation phase. This enables the designer to express preferences relative to each objective function with more detail. Firstly, PP translates the designer's knowledge into classes<sup>6</sup> with previously defined ranges.<sup>7</sup> This preference set reveals the DM's wishes using physical units for each of the objectives in the MOP. From this point of view, the problem is moved to a different range where all the variables are independent of the original MOP (see Figure 4).

For each objective and set of preferences  $\mathfrak{P}$ , a class function  $\eta_q(\mathbf{J}(\boldsymbol{\theta}))|_{\mathfrak{P}}$ ,  $q = [1, \dots, m]$  is built to translate each objective  $J_q(\boldsymbol{\theta})$  to a new range where all the objectives are equivalent to each other. A PP index  $J_{pp}(\mathbf{J}(\boldsymbol{\theta})) = \sum_{q=1}^m \eta_q(\mathbf{J}(\boldsymbol{\theta}))$  is then calculated.

For example, in Figure 4, two objectives  $J_1(\boldsymbol{\theta})$  and  $J_2(\boldsymbol{\theta})$  are optimised (in both cases: *the smaller, the better*). At this point, there is not enough information to compare two objective values  $J_1(\boldsymbol{\theta}) = a$  against  $J_2(\boldsymbol{\theta}) = b$ ; nevertheless, using a preference set, it is possible to map each objective to a certain range of preferences and so enable the comparison. In this example  $J_1(\boldsymbol{\theta}) = a$  is in the undesirable region (U) and  $J_2(\boldsymbol{\theta}) = b$  is in the tolerable zone (T). Therefore, it is possible to compare both objective values  $a$  and  $b$ :  $b$  is *less desirable* than  $a$ ; by assigning numerical values in this new range, it is possible for an algorithm interpreting these semantic values to perform a preference based optimisation. This information is exploited by the optimiser, for example, by means of the OVO (*one-vs-others*) rule of the original PP technique: some degradation will be accepted in the objective  $J_2(\boldsymbol{\theta})$  if that implies a better

---

<sup>6</sup>The original method states 4 classes: 1S (smaller is better); 2S (larger is better); 3S (a value is better); and 4S (a range is better)

<sup>7</sup>According to the original method: highly desirable (HD), desirable (D), tolerable (T), undesirable (U) and highly undesirable (HU)



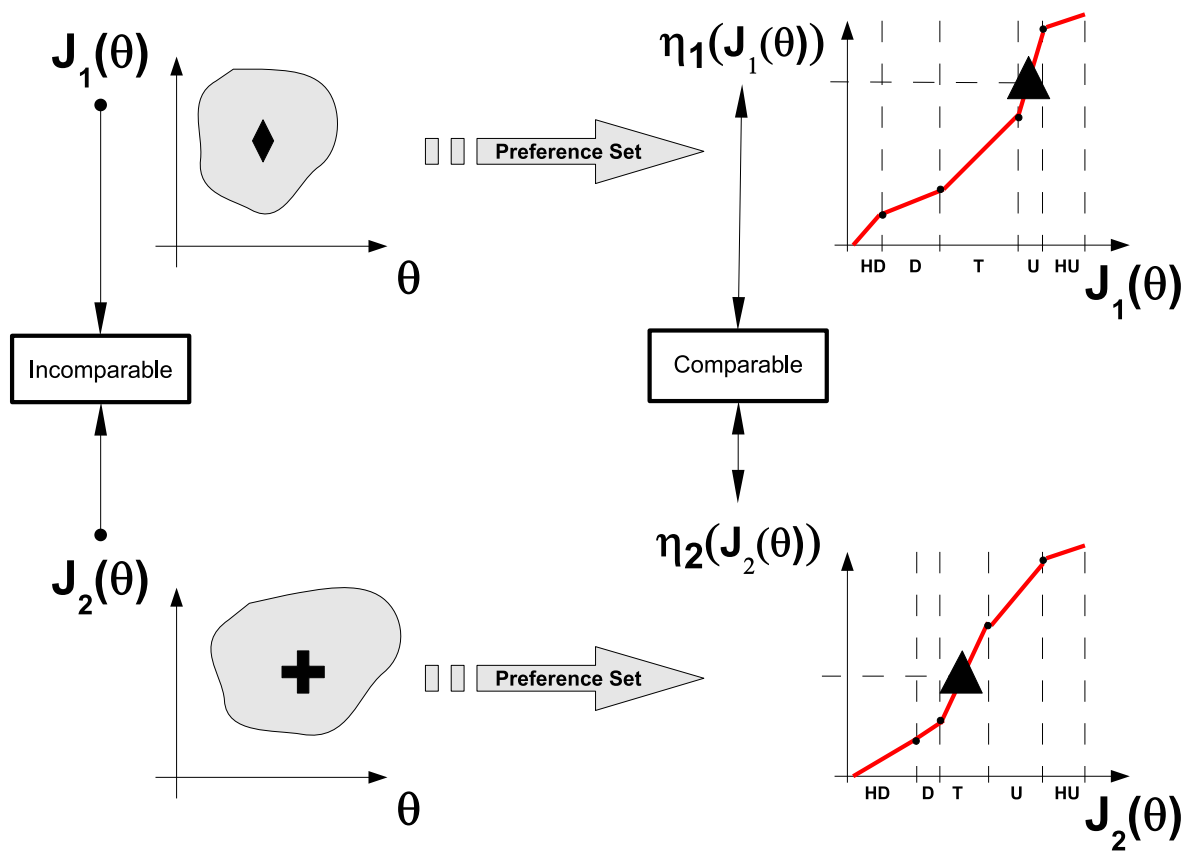


Figure 4: Physical programming (PP) notion. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

performance of  $J_1(\boldsymbol{\theta})$  with the aim of placing both values in the tolerable region. This is achieved by optimizing the  $J_{pp}(\mathbf{J}(\boldsymbol{\theta})) = \eta_1(J_1(\boldsymbol{\theta})) + \eta_2(J_2(\boldsymbol{\theta}))$  index.

In this work, the implementation stated in [48], namely Global Physical Programming (GPP), is a better fit for evolutionary optimisation techniques and will be used. This is due to the fact that the original method employs several resources to build the proper class functions  $\eta_q(\mathbf{J}(\boldsymbol{\theta}))|_{\mathfrak{P}}$ , to fulfill a list of convexity and continuity requirements. The interested reader might refer to [48, 49] for a detailed explanation. For the sake of simplicity such details are not reproduced here, since they will be used as a basis for the development to be presented in Section 3.

The  $J_{pp}(\mathbf{J}(\boldsymbol{\theta}))$  index is suitable to evaluate the performance of a design alternative, but not that of the design concept. That is, if it is used as it is, it will evolve the entire population to a single Pareto optimal solution. Therefore, it must be merged with other mechanisms to maintain diversity in the Pareto front. Pruning mechanisms seem to be a promising solution for this purpose. Therefore, GPP will be used with the spMODE algorithm (see Algorithm 4 in the next section and references [50, 51]), which is a MOEA based on the differential evolution algorithm [52, 53, 54] and spherical coordinates to prune  $\mathbf{J}_P^*$ . Even if similar algorithms use similar approaches [55, 56, 57], the usage of a norm to perform the pruning makes it suitable to incorporate preferences, as detailed below.

#### 2.4. Spherical pruning review

A general pseudocode for MOEAs with pruning mechanism and external archive  $A$  is shown in Algorithm 1. The usage of an external archive  $A$  to store the best set of quality solutions found so far in an evolutionary process is common in EMO.

The basic idea of the spherical pruning is to analyze the proposed solutions in the current Pareto front approximation  $\mathbf{J}_P^*$  by using normalized spherical coordinates from a reference solution (see Figure 5). With such an approach, it is possible to attain a good distribution along the Pareto front [50, 51]. The algorithm selects one solution for each spherical sector (see Figure 5), according to a given norm or measure. This process is explained in Algorithm 2, where the following definitions are required:

**Definition 4.** (*normalized spherical coordinates*) given a solution  $\boldsymbol{\theta}^1$  and  $\mathbf{J}(\boldsymbol{\theta}^1)$ , let

$$\mathbf{S}(\mathbf{J}(\boldsymbol{\theta}^1)) = [\|\mathbf{J}(\boldsymbol{\theta}^1)\|_2, \boldsymbol{\beta}(\mathbf{J}(\boldsymbol{\theta}^1))] \quad (5)$$

be the normalized spherical coordinates from a reference solution  $\mathbf{J}^{ref}$  where  $\boldsymbol{\beta}(\mathbf{J}(\boldsymbol{\theta}^1)) = [\beta_1(\mathbf{J}(\boldsymbol{\theta}^1)), \dots, \beta_{m-1}(\mathbf{J}(\boldsymbol{\theta}^1))]$  is the arc vector and  $\|\mathbf{J}(\boldsymbol{\theta}^1)\|_2$  the Euclidean distance to the reference solution.

It is important to guarantee that  $\mathbf{J}^{ref}$  dominates all the solutions. An intuitive approach is to select:

$$\mathbf{J}^{ref} = \mathbf{J}^{ideal} = [\min J_1(\boldsymbol{\theta}^i), \dots, \min J_m(\boldsymbol{\theta}^i)] \forall \mathbf{J}(\boldsymbol{\theta}^i) \in \hat{A}|_G \quad (6)$$

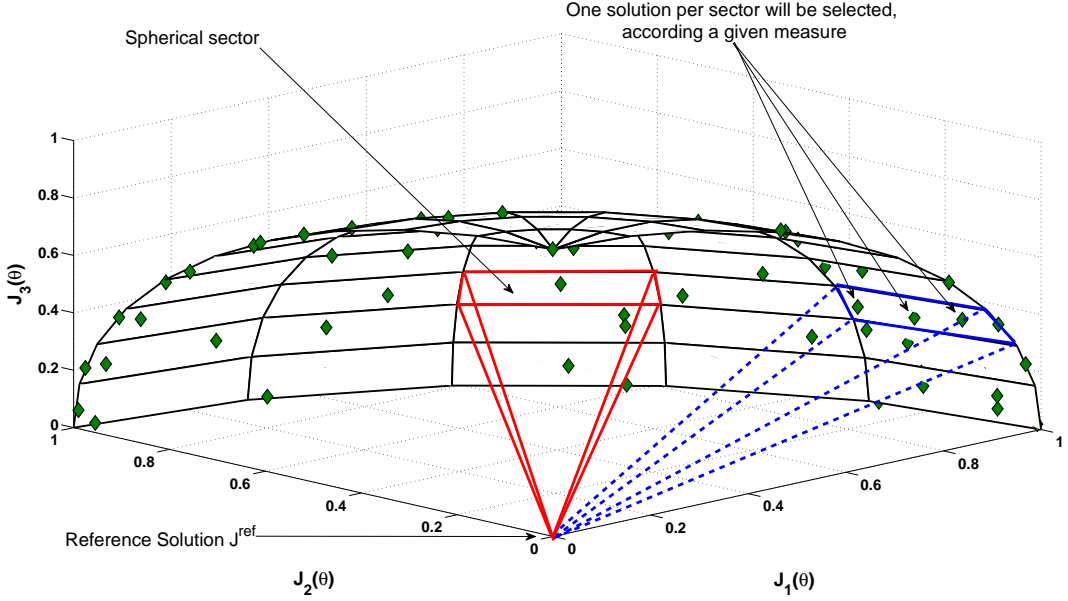


Figure 5: Spherical relations on  $\mathbf{J}_P^* \subset \mathbb{R}^3$ . For each spherical sector, just one solution, the solution with the lowest norm will be selected.

**Definition 5.** (*sight range*) The sight range from the reference solution  $\mathbf{J}^{ref}$  to the Pareto front approximation  $\mathbf{J}_P^*$  is bounded by  $\beta^U$  and  $\beta^L$ :

$$\beta^U = [\max \beta_1(\mathbf{J}(\theta^i)), \dots, \min \beta_{m-1}(\mathbf{J}(\theta^i))] \quad \forall \mathbf{J}(\theta^i) \in \hat{A}|_G \quad (7)$$

$$\beta^L = [\min \beta_1(\mathbf{J}(\theta^i)), \dots, \min \beta_{m-1}(\mathbf{J}(\theta^i))] \quad \forall \mathbf{J}(\theta^i) \in \hat{A}|_G \quad (8)$$

If  $\mathbf{J}^{ref} = \mathbf{J}^{ideal}$ , it is straightforward to prove that  $\beta^U = [\frac{\pi}{2}, \dots, \frac{\pi}{2}]$  and  $\beta^L = [0, \dots, 0]$ .

**Definition 6.** (*spherical grid*) Given a set of solutions in the objective space, the spherical grid on the  $m$ -dimensional space in arc increments  $\beta_\epsilon = [\beta_1^\epsilon, \dots, \beta_{m-1}^\epsilon]$  is defined as:

$$\Lambda^{\mathbf{J}_P^*} = \left[ \frac{\beta_1^U - \beta_1^L}{\beta_1^\epsilon}, \dots, \frac{\beta_{m-1}^U - \beta_{m-1}^L}{\beta_{m-1}^\epsilon} \right] \quad (9)$$

**Definition 7.** (*spherical sector*) The normalized spherical sector of a solution  $\theta^1$  is defined as:

$$\Lambda_\epsilon(\theta^1) = \left[ \left[ \frac{\beta_1(\mathbf{J}(\theta^1))}{\Lambda_1^{\mathbf{J}_P^*}} \right], \dots, \left[ \frac{\beta_{m-1}(\mathbf{J}(\theta^1))}{\Lambda_{m-1}^{\mathbf{J}_P^*}} \right] \right] \quad (10)$$

**Definition 8.** (*spherical pruning*) given two solutions  $\theta^1$  and  $\theta^2$  from a set,  $\theta^1$  has preference in the spherical sector over  $\theta^2$  if:

$$[\Lambda_\epsilon(\theta^1) = \Lambda_\epsilon(\theta^2)] \wedge [\|\mathbf{J}(\theta^1)\|_p < \|\mathbf{J}(\theta^2)\|_p] \quad (11)$$

1	Generate initial population $P _0$ with $N_p$ individuals;
2	Evaluate $P _0$ ;
3	Apply dominance criterion on $P _0$ to obtain archive $A _0$ ;
4	<b>while</b> <i>stopping criterion unsatisfied</i> <b>do</b>
5	Update generation counter $G=G+1$ ;
6	Obtain subpopulation $S _G$ with solutions in $P _G$ and $A _G$ ;
7	Generate offspring $O _G$ with $S _G$ ;
8	Evaluate offspring $O _G$ ;
9	Update population $P _G$ with offspring $O _G$ ;
10	Apply dominance criterion on $O _G \cup A _G$ to obtain $\hat{A} _G$ ;
11	Apply pruning mechanism to prune $\hat{A} _G$ to obtain $A _{G+1}$ ;
12	Update environment variables (if using a self-adaptive mechanism);
13	<b>end</b>
14	Algorithm terminates. $\mathbf{J}_P$ is approximated by $\mathbf{J}_P^* = A _G$ ;

**Algorithm 1:** MOEA with pruning mechanism

where  $\|\mathbf{J}(\boldsymbol{\theta})\|_p = \left( \sum_{q=1}^m |J_q(\boldsymbol{\theta})|^p \right)^{1/p}$  is a suitable  $p$ -norm.

In this implementation, spherical pruning is merged with the DE algorithm [52, 54, 58]. Although any other evolutionary or nature-inspired mechanism may be used, the DE algorithm was selected because of its simplicity. The DE algorithm uses two operators: mutation and crossover (Equations (12) and (13) respectively) to generate its offspring (Algorithm 3).

**Mutation:** For each target (parent) vector  $\boldsymbol{\theta}^i|_G$ , a mutant vector  $\mathbf{v}^i|_G$  is generated at generation  $G$  according to Equation (12):

$$\mathbf{v}^i|_G = \boldsymbol{\theta}^{r_1}|_G + F(\boldsymbol{\theta}^{r_2}|_G - \boldsymbol{\theta}^{r_3}|_G) \quad (12)$$

Where  $r_1 \neq r_2 \neq r_3 \neq i$  and  $F$  is usually known as the scaling factor.

**Crossover:** For each target vector  $\boldsymbol{\theta}^i|_G$  and its mutant vector  $\mathbf{v}^i|_G$ , a trial (child) vector  $\mathbf{u}^i|_G = [u_1^i|_G, u_2^i|_G, \dots, u_n^i|_G]$  is created as follows:

$$u_j^i|_G = \begin{cases} v_j^i|_G & \text{if } \text{rand}(0, 1) \leq Cr \\ \theta_j^i|_G & \text{otherwise} \end{cases} \quad (13)$$

where  $j \in 1, 2, 3 \dots n$  and  $Cr$  is named the crossover probability rate.

The standard selection mechanisms are as follows:

- For single objective optimisation, a child is selected over its parent (for the next generation) if it has a better cost value.

```

1 Read archive  $\hat{A}|_G$ ;
2 Read and update extreme values for  $J_{ref}|_G$ ;
3 for each member in  $\hat{A}|_G$  do
4 | calculate its normalized spherical coordinates (Definition 3);
5 end
6 Build the spherical grid (Definition 4 and 5);
7 for each member in  $\hat{A}|_G$  do
8 | calculate its spherical sector (Definition 6);
9 end
10 for  $i=1:SolutionsInArchive$  do
11 | Compare with the remainder solutions in  $\hat{A}|_G$ ;
12 | if no other solution has the same spherical sector then
13 | | it goes to archive  $A|_{G+1}$ ;
14 | end
15 | if other solutions are in the same spherical sector then
16 | | it goes to archive  $A|_{G+1}$  if it has the lowest norm (Definition 7);
17 | end
18 end
19 Return Archive  $A|_{G+1}$ ;

```

**Algorithm 2:** Spherical pruning mechanism

```

1 for  $i=1:SolutionsInParentPopulation$  do
2 | Generate a Mutant Vector  $\mathbf{v}^i$  (Equation (12)) ;
3 | Generate a Child Vector  $\mathbf{u}^i$  (Equation (13)) ;
4 end
5 Offspring  $O = U$ ;

```

**Algorithm 3:** DE offspring generation mechanism

- In EMO, a simple selection based on dominance is used; a child is selected over his parent if the child dominates his parent (Definition 3).

```

1 Generate initial population  $P|_0$  with  $N_p$  individuals;
2 Evaluate  $P|_0$ ;
3 Apply dominance criterion on  $P|_0$  to obtain  $A|_0$ ;
4 while stopping criterion unsatisfied do
5   Read generation count  $G$ ;
6   Obtain subpopulation  $S|_G$  with solutions in  $P|_G$  and  $A|_G$ ;
7   Generate offspring  $O|_G$  with  $S|_G$  using DE operators (Algorithm 3).;
8   Evaluate offspring  $O|_G$ ;
9   Update population  $P|_G$  with offspring  $O|_G$  according to greedy selection
   mechanism.;
10  Apply dominance criterion on  $O|_G \cup A|_G$  to obtain  $\hat{A}|_G$ ;
11  Apply pruning mechanism based on spherical pruning (Algorithm 2) to prune  $\hat{A}|_G$ 
   to obtain  $A|_{G+1}$ ;
12   $G = G + 1$ ;
13 end
14 Algorithm terminates.  $\mathbf{J}_P$  is approximated by  $\mathbf{J}_P^* = A|_G$ ;

```

**Algorithm 4:** spMODE.

This solution, merging the DE algorithm and the spherical pruning mechanism (Algorithm 2) was termed the spMODE algorithm (see Algorithm 4) and it is freely available at Matlab© Central<sup>8</sup>. At this point, we will present the preference handling proposal of this paper.

### 3. Pertinency Improvement by means of GPP

Global physical programming is a tool that can be used in different ways by MOEAs. In this case, it will be merged together with a pruning technique, in order to decide which solutions will be archived in an external file. GPP can be used as a selection mechanism in the evolved population and/or in the store and replace the mechanism in the external archive  $A$ .

#### 3.1. Global physical programming statements

Given a vector  $\varphi \in \mathbb{R}^m$ , linear functions will be used for class functions  $\eta_q(\varphi)|_{\mathfrak{F}}$  as detailed in [48]<sup>9</sup>, due to their simplicity and interpretability. Firstly, an offset between two

---

<sup>8</sup><http://www.mathworks.com/matlabcentral/fileexchange/39215>

<sup>9</sup>Hereafter, only 1S classes (the smaller, the better) will be considered.

adjacent ranges is incorporated (see Figure 6) to meet the OVO rule criterion [59, 60]. Given a set of preferences  $\mathfrak{P}$  with  $M$  ranges for  $m$  objectives:

$$\mathfrak{P} = \begin{pmatrix} J_1^1 & \cdots & J_1^M \\ \vdots & \ddots & \vdots \\ J_m^1 & \cdots & J_m^M \end{pmatrix} \quad (14)$$

$\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}}$ ,  $q = [1, \dots, m]$  are defined as:

$$\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}} = \alpha_{k-1} + \delta_{k-1} + \Delta\alpha_k \frac{\varphi_q - J_q^{k-1}}{J_q^k - J_q^{k-1}} \quad (15)$$

$$J_q^{k-1} \leq \varphi_q < J_q^k \quad (16)$$

where

$$\alpha_0 = 0 \quad (17)$$

$$\alpha_1 \in \mathbb{R}^+ \quad (18)$$

$$\alpha_k > \alpha_{k-1} \quad (1 < k \leq M) \quad (19)$$

$$\Delta\alpha_k = \alpha_k - \alpha_{k-1} \quad (1 \leq k \leq M) \quad (20)$$

$$\delta_0 = 0 \quad (21)$$

$$\delta_1 \in \mathbb{R}^+ \quad (22)$$

$$\delta_k > m \cdot (\alpha_k + \delta_{k-1}) \quad (1 < k \leq M) \quad (23)$$

The last inequality guarantees the OVO rule, since an objective value in a given range is always greater than the sum of the others in a more preferable range. Therefore, the  $J_{gpp}(\boldsymbol{\varphi})$  index is defined as:

$$J_{gpp}(\boldsymbol{\varphi}) = \sum_{q=1}^m \eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}} \quad (24)$$

The  $J_{gpp}(\boldsymbol{\varphi})$  has an intrinsic structure to deal with constraints. If the fulfillment of constraints is required, they will be included in the preference set as objectives. That is, preference ranges will be stated for each constraint and they will be used to compute the  $J_{gpp}(\boldsymbol{\varphi})$  index. The  $\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}}$  is shown in Figure 6 for the specific case (to be used hereafter) of the following 5 preference ranges:

**HD:** Highly desirable if  $J_q^0 \leq J_q(\boldsymbol{\varphi}) < J_q^1$ .

**D:** Desirable if  $J_q^1 \leq J_q(\boldsymbol{\varphi}) < J_q^2$ .

**T:** Tolerable if  $J_q^2 \leq J_q(\boldsymbol{\varphi}) < J_q^3$ .

**U:** Undesirable if  $J_q^3 \leq J_q(\varphi) < J_q^4$ .

**HU:** Highly undesirable  $J_q^4 \leq J_q(\varphi) < J_q^5$ .

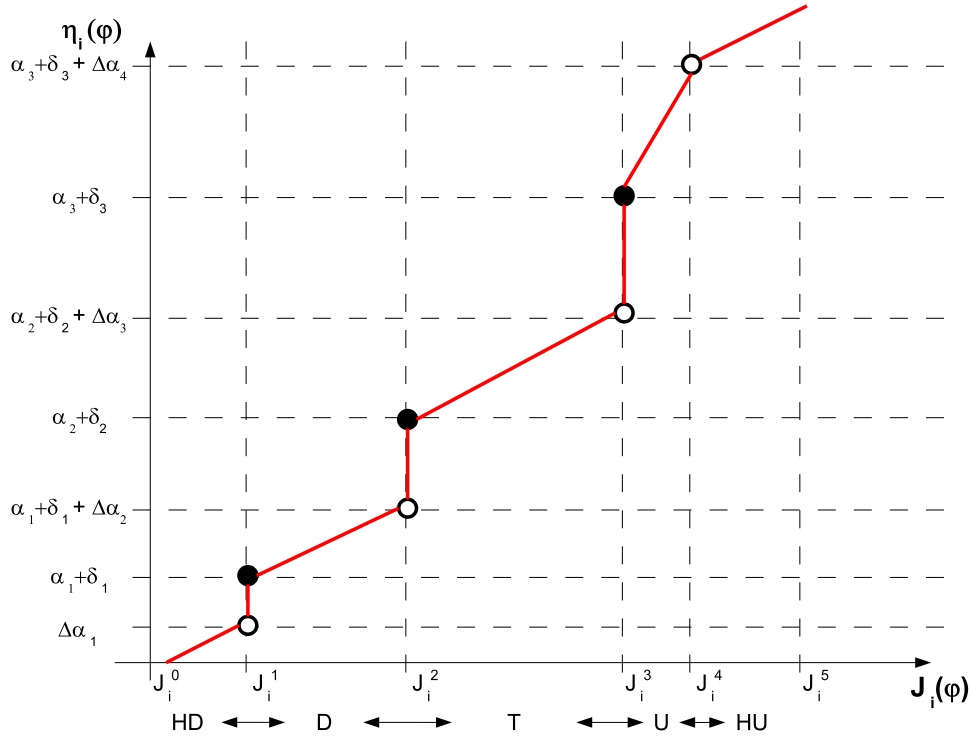


Figure 6: New class definition for global physical programming.

Those preference ranges are defined for the sake of flexibility (as it will be shown) to evolve the population to a pertinent Pareto front. The following definitions will be used (see Figure 7):

**T\_Vector:**  $\mathbf{J}^T = [J_1^3, J_2^3, \dots, J_m^3]$ , *i.e.* the vector with the maximum value for each objective in the tolerable range.

**D\_Vector:**  $\mathbf{J}^D = [J_1^2, J_2^2, \dots, J_m^2]$ , *i.e.* the vector with the maximum value for each objective in the desirable range.

**HD\_Vector:**  $\mathbf{J}^{HD} = [J_1^1, J_2^1, \dots, J_m^1]$ , *i.e.* the vector with the maximum value for each objective in the highly desirable range.

**T\_HypV:** The hypervolume of the Pareto front approximation bounded by  $\mathbf{J}^T$ .

**D\_Hypv:** The hypervolume of the Pareto front approximation bounded by  $\mathbf{J}^D$ .

**HD\_HypV:** The hypervolume of the Pareto front approximation bounded by  $\mathbf{J}^{HD}$ .



$\mathbf{T}_P^*$ : the tolerable Pareto front approximation where all solutions dominate  $\mathbf{J}^T$ .

$\mathbf{D}_P^*$ : the desirable Pareto front approximation where all solutions dominate  $\mathbf{J}^D$ .

$\mathbf{HD}_P^*$ : the highly desirable Pareto front approximation where all solutions dominate  $\mathbf{J}^{HD}$ .

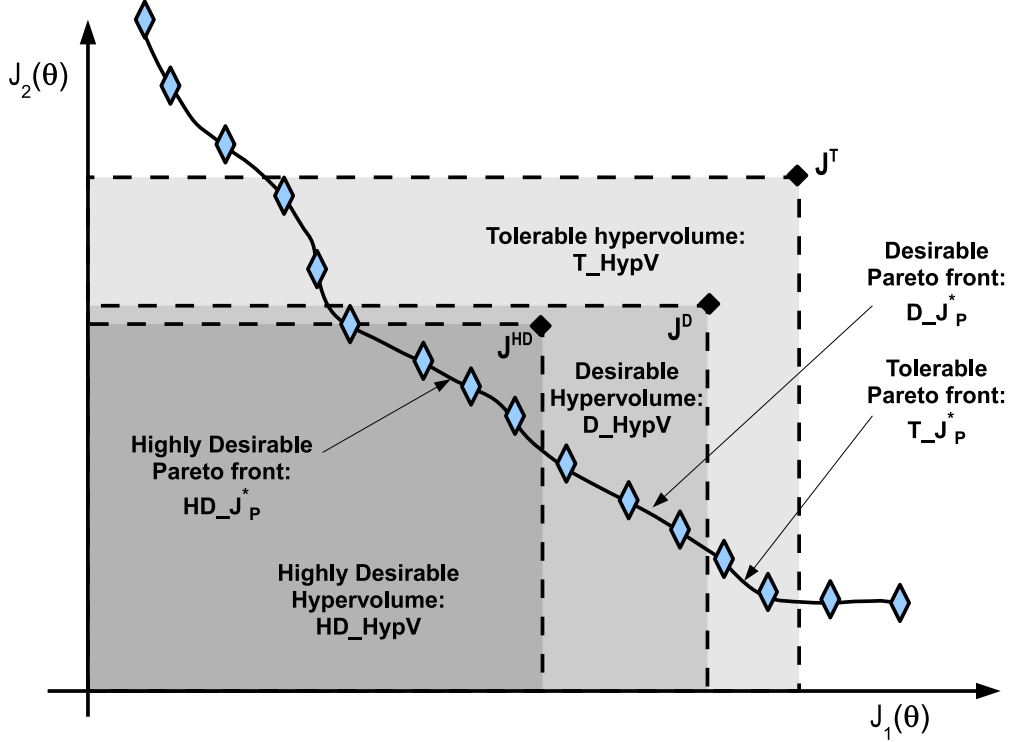


Figure 7: Graphical representation of the definitions stated.

### 3.2. Population selection and archive update

The  $J_{gpp}(\varphi)$  index will be used as a selection mechanism in the evolutionary technique. Nevertheless, using it through the entire evolution process is not a practical approach. This is because the  $J_{gpp}(\varphi)$  would lead the entire population to converge to a single solution, since the physical index converges to a single Pareto optimal solution. To avoid this, a mechanism must be designed to evolve the population to a zone of interest and then encourage diversity. In this case, a switch operator is used in DE to change the selection criteria (Algorithm 5) at a certain value  $J_{gpp}^{max}$ .

This  $J_{gpp}^{max}$  value needs to be previously defined. This upper bound on  $J_{gpp}(\varphi)$  will push the population to evolve to a desired preference region. As five preference ranges are defined, an intuitive selection of such a value is  $J_{gpp}^{max} = J_{gpp}(\mathbf{J}^T)$ . This will guarantee the population evolves into the feasible **T\_HypV** and then performs a selection based on dominance. The evolutionary process has a strong pressure to reach the **T\_HypV** (*dominance is essential*);

furthermore, it is guaranteed that only tolerable Pareto optimal solutions will be contained in  $\mathbf{J}_P^*$ .

In the case of the archiving strategy to update  $A$ ,  $J_{gpp}(\boldsymbol{\varphi})$  is used as a pseudo-norm measurement to select one solution for each spherical sector (see Algorithm 6). With this, the *most preferable solution* according to the set of preferences  $\mathfrak{P}$  previously defined by the designer will be retained in each spherical sector.

```

1 Read generation counter  $G$ ;
2 Read offspring (child population)  $O|_G$  and subpopulation (parent)  $S|_G$ ;
3 for  $i=1$ :Solutions in child population do
4   Calculate the physical index  $J_{gpp}(\mathbf{J}(\mathbf{u}^i))$  and  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}^i))$ ;
5   if  $J_{gpp}(\mathbf{J}(\mathbf{u}^i)) > J_{gpp}^{max}$  AND  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}^i)) > J_{gpp}^{max}$  then
6     if  $J_{gpp}(\mathbf{J}(\mathbf{u}^i)) < J_{gpp}(\mathbf{J}(\boldsymbol{\theta}^i))$  then
7        $\mathbf{u}^i$  substitutes  $\boldsymbol{\theta}^i$  in the main population  $P|_G$ 
8     end
9   end
10  if  $J_{gpp}(\mathbf{J}(\mathbf{u}^i)) < J_{gpp}^{max}$  AND  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}^i)) > J_{gpp}^{max}$  then
11     $\mathbf{u}^i$  substitutes  $\boldsymbol{\theta}^i$  in the main population  $P|_G$ 
12  end
13  if  $J_{gpp}(\mathbf{J}(\mathbf{u}^i)) < J_{gpp}^{max}$  AND  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}^i)) < J_{gpp}^{max}$  then
14    if  $\mathbf{u}^i \preceq \boldsymbol{\theta}^i$  AND  $\boldsymbol{\theta}^i \in P|_G$  then
15       $\mathbf{u}^i$  substitutes  $\boldsymbol{\theta}^i$  in the main population  $P|_G$ 
16    end
17  end
18 end
19 Selection procedure terminates. Return parent population  $P|_G$ ;

```

**Algorithm 5:** DE selection procedure with global physical programming.

### 3.3. Tolerable solution handling

The previously mentioned feature could be enough if the DM is satisfied with considering solutions that might have all their individual objective values in the tolerable region (for example). Nevertheless, the DM is usually willing to accept just *some* of the objectives in a given region (see Figure 8). Such a feature can be incorporated into the pruning mechanism by modifying  $J_{gpp}^{max}$ . For example, it is assumed that the DM is dealing with an  $m$ -objective problem. The following values<sup>10</sup> could be stated for  $J_{gpp}^{max}$ :

- If  $J_{gpp}^{max} = J_{gpp}(\mathbf{J}^T)$ , then solutions with their  $m$  objectives in the tolerable region might appear in  $\mathbf{J}_P^*$ .

<sup>10</sup>without loss of generality, values are modified starting from objective  $m$  towards objective 1.

```

1 Read generation counter G;
2 Read  $\hat{A}|_G$  to be prune;
3 Read and update extreme values for  $J_{ref}|_G$ ;
4 for each member in  $\hat{A}|_G$  do
5   | calculate its normalized spherical coordinates (Definition 3)
6 end
7 Build the spherical grid (Definitions 4 and 5);
8 for each member in  $\hat{A}|_G$  do
9   | calculate its spherical sector (Definition 6)
10 end
11 for  $i=1:SolutionsInParentPopulation$  do
12   | if  $J_{gpp}(\mathbf{J}(\theta^i)) > J_{gpp}^{max}$  then
13     |  $\theta^i$  is discarded
14   | end
15   | if  $J_{gpp}(\mathbf{J}(\theta^i)) \leq J_{gpp}^{max}$  then
16     | Compare with the remaining solutions in  $A|_G$ ;
17     | if no other solution has the same spherical sector then
18       | it goes to the archive  $A|_{G+1}$ 
19     | end
20     | if other solutions are in the same spherical sector then
21       | it goes to the archive  $A|_{G+1}$  if it has the lowest  $J_{gpp}(\mathbf{J}(\theta^i))$ 
22     | end
23   | end
24 end
25 Pruning ends. Return  $A|_{G+1}$ ;

```

**Algorithm 6:** Spherical pruning with physical programming index.

- If  $J_{gpp}^{max} = J_{gpp}(\overbrace{[J_1^3, J_2^3, \dots, J_{m-1}^3]}^{Tolerable}, \overbrace{J_m^2}^{Desirable})$ , then solutions with their  $m$  objectives in the tolerable region will not appear in  $\mathbf{J}_P^*$ .
- If  $J_{gpp}^{max} = J_{gpp}(\overbrace{[J_1^3, J_2^3, \dots, J_{m-2}^3]}^{Tolerable}, \overbrace{[J_{m-1}^2, J_m^2]}^{Desirable})$ , then solutions with  $m - 1$  or more objectives in the tolerable region will not appear in  $\mathbf{J}_P^*$ .
- If  $J_{gpp}^{max} = J_{gpp}(\overbrace{[J_1^3, J_2^3, \dots, J_{m-3}^3]}^{Tolerable}, \overbrace{[J_{m-2}^2, J_{m-1}^2, J_m^2]}^{Desirable})$ , then solutions with  $m - 2$  or more objectives in the tolerable region will not appear in  $\mathbf{J}_P^*$ .
- And so on.

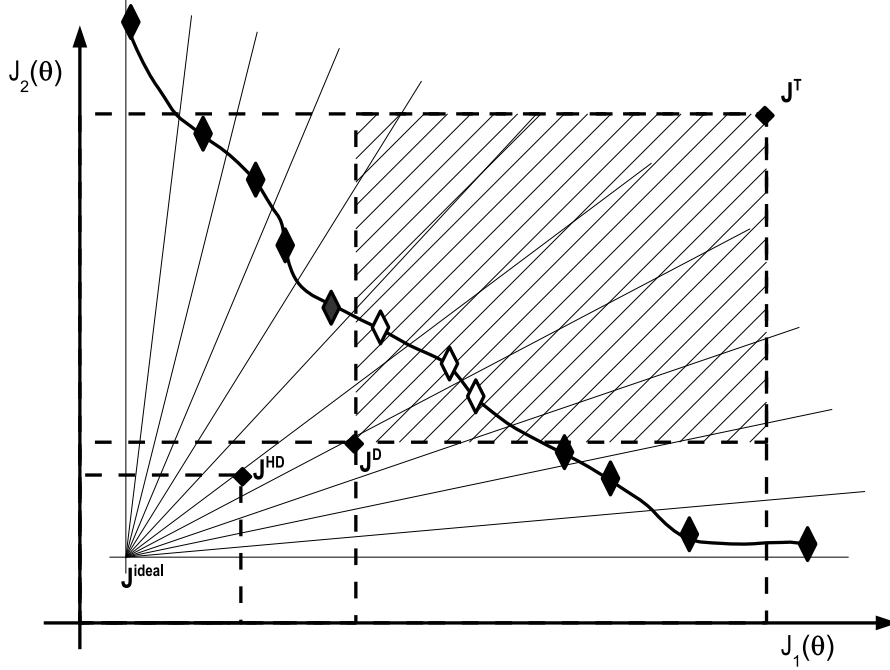


Figure 8: Handling of tolerable solutions. The algorithm will avoid (on the designer's request) solutions with several tolerable values (light solutions) according with the  $J_{gpp}^{max}$  value defined. In the example, bi-objective vectors with both values in the tolerable zone are omitted.

### 3.4. Multiple preference coding

The designer may state more than one preference set  $\mathfrak{P}$  for a given MOP. This could be the case of a many-objective optimisation, where the DM is willing to accept some degradation in one objective, if he or she can assure outstanding performance on the remainder (see Figure 9). It is assumed that the designer defines  $K$  different preference sets; therefore, the Equation (24) is redefined as:

$$J_{gpp}(\varphi) = \min_{k=[1, \dots, K]} \left( \sum_{q=1}^m \eta_q(\varphi) |_{\mathfrak{P}_k} \right) \quad (25)$$

### 3.5. Pareto front approximation size control

The GPP can be used as a mechanism to dynamically adapt the algorithm according to the desired set of solutions. Instead of reducing the grid size,  $J_{gpp}^{max}$  will be modified to retain the most preferable solutions (Figure 10). A threshold for a desired number of solutions  $[\underline{car}(J_P^*), \overline{car}(J_P^*)]$  will be stated. A simple dynamic adaptation of  $J_{gpp}^{max}$  is described in Algorithm 7. With this mechanism, the pruning technique will concentrate towards the **HD-HypV**, if the required number of solutions in the **T-HypV** is beyond the bound  $\underline{car}(J_P^*)$ .

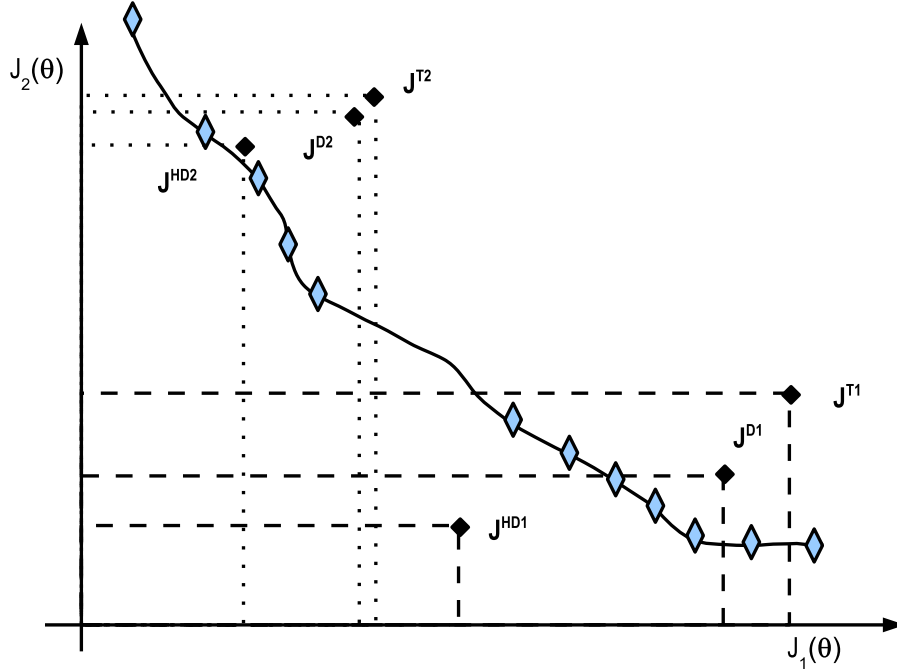


Figure 9: Multiple preference set definition.

```

1 if  $car(A|_G) > \overline{car(J_P^*)}$  then
2   | Sort elements on  $A|_G$  according to their  $J_{gpp}(\mathbf{J}(\theta))$  index;
3   | Substitute  $J_{gpp}^{max}$  value with the physical norm of the  $\overline{car(J_P^*)}$ -th element;
4   | Prune any element on  $A|_G$  with  $J_{gpp}(\mathbf{J}(\theta)) > J_{gpp}^{max}$ 
5 end
6 Return  $A|_G$ ;

```

**Algorithm 7:** Dynamic size control.

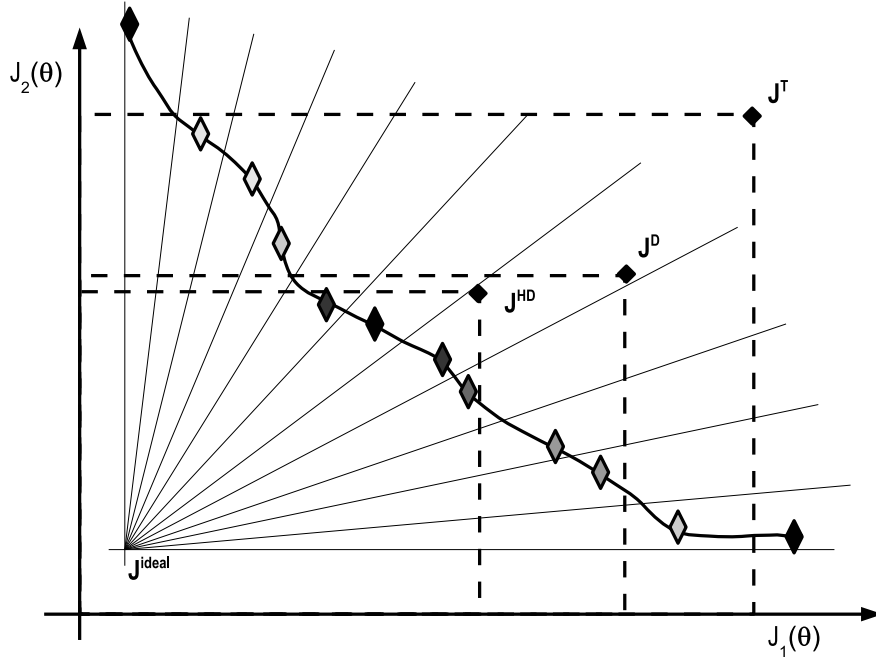


Figure 10: Dynamic size control. Less preferable solutions are omitted (on the designer’s request) in the external archive  $A$ . The darker the solution, the more valuable according to the preference set. Extreme solutions are always preserved.

### 3.6. Algorithm proposal: *spMODE-II*.

With the aforementioned Algorithms 3, 5, 6 and 7, it is possible to re-write Algorithms 1 and 2 in order to state a proposal using GPP to improve pertinency of the  $\mathbf{J}_P^*$ . Therefore, a preference based spherical pruning with multi-objective differential evolution algorithm (*spMODE-II*) preferences is presented (see Algorithm 8). Some discussions on this proposal follows below.

#### 3.6.1. Comparison with other approaches: strengths and limitations

As commented in the introduction, the PP technique has been merged before with multi-objective optimization algorithms. For example, in [1], an aggregate objective function approach was used, running several optimization statements to generate a Pareto front approximation. Differences with this approach (beyond the structural differences of using a MOEA instead of a local optimisation algorithm), are related with the features commented on previously. While this algorithm does not incorporate a mechanism for multiple preference conditions or size control, the *spMODE-II* does.

According to the classification presented in Section 2.2, the *spMODE-II* algorithm uses an *a priori* preference handling mechanism. Such a mechanism, based on GPP, states a framework where *dominance is essential* (see Algorithm 5), and enables the algorithm to define priorities on design alternatives according to an *objective value against objective value* basis (see Algorithm 6). This algorithm is capable of handling multiple preference conditions

```

1 Generate initial population  $P|_0$  with  $N_p$  individuals;
2 Evaluate  $P|_0$ ;
3 Apply dominance criterion on  $P|_0$  to obtain  $A|_0$ ;
4 while stopping criterion not satisfied do
5     Read generation count  $G$ ;
6     Obtain subpopulation  $S|_G$  with solutions in  $P|_G$  and  $A|_G$ ;
7     Generate offspring  $O|_G$  with  $S|_G$  using DE operators (Algorithm 3).;
8     Evaluate offspring  $O|_G$ ;
9     Update population  $P|_G$  with offspring  $O|_G$  according to  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}))$  values
    (Algorithm 5);
10    Apply dominance criterion on  $O|_G \cup A|_G$  to obtain  $\hat{A}|_G$ ;
11    Apply pruning mechanism based on  $J_{gpp}(\mathbf{J}(\boldsymbol{\theta}))$  (Algorithm 6) to prune  $\hat{A}|_G$  to
    obtain  $A|_{G+1}$ ;
12    Apply dynamic size control on  $A|_{G+1}$  (Algorithm 7);
13    Update environment variables (if using a self-adaptive mechanism);
14     $G = G + 1$ ;
15 end
16 Algorithm terminates.  $\mathbf{J}_P$  is approximated by  $\mathbf{J}_P^* = A|_G$ ;

```

**Algorithm 8:** spMODE-II.

simultaneously (which could be helpful to deal with *subset against subset* priorities) and approximating Pareto optimal solutions for each of them (see Equation 25); it is indifferent to the Pareto front shape, and enables the designer to specify how many solutions are required for the MCDM step (see Algorithm 7).

Mechanisms such as goal vectors, which use one or more proposal vectors to redirect the evolution process, are similar to the spMODE-II algorithm, in the sense that *dominance is essential*. Nevertheless, such vectors need to be carefully chosen. A goal vector selected inside or outside of the Pareto front could affect the algorithm performance. The presented proposal brings the flexibility of a better redirection of the evolution process.

The proposal also assumes that the designer has an idea about the undesirability, tolerability, or desirability for a given objective value. That may not be the case, and sometimes for the designer it is only important to prioritize some of the objectives (*Objective against objective*). This is a limitation for the algorithm; therefore, in such a case, fitter mechanisms such as the one proposed in [61] should be used.

Finally, another limitation of this approach is that it states the preferences only in an *a priori* step. In the event that the designer is willing to refine a given set(s) of preferences *on the fly* (*progressive* scheme), then proposals such as [62] will be more appropriate for such situations.

### 3.6.2. Using spMODE-II in EMO: discussions and insights

Using GPP or related approaches brings the additional task of defining  $K$  preference sets  $\mathfrak{P}$ . Nevertheless, in several instances this supplementary effort could be justifiable if it brings a Pareto set approximation with more pertinent solutions to the DM, and so facilitating the decision making step. Therefore, the DM must be willing to spend this additional effort at the beginning of the MOP statement definition. If upper and lower bounds on objectives are known and are sufficient to improve pertinency, a simple constraint coding could be used, such as the one proposed in [4] where bounds on  $\mathbf{J}_P^*$  are stated.

A statement to discourage the usage of the approach presented here could be the need to define the preference set  $\mathfrak{P}$ . It is fundamental to have an understanding of the objectives to define the preference ranges. Nevertheless, if the DM has no idea about such values, it could be an indicative of a perfunctory or precipitate selection of design objectives. Therefore, perhaps the DM should ponder the design objectives stated. One of the advantages of MOEAs is the use of more interpretable and meaningful objectives for the DM and this aspect should be exploited.

Finally, according to the number of solutions, the designer could adopt a *ten times the number of objectives* rule of thumb, based on [31]. In such work, it is noticed that for 2-3 objectives a set of 20-30 design alternatives could be handled by the DM. In Table 1 guidelines for parameter adjustment of the spMODE-II algorithm are given.

## 4. Experimental Setup

Different tests are defined to evaluate the presented proposal. In all cases, parameters used for DE are  $F = 0.5$  and  $Cr = 0.9$  with an initial population of 50 individuals. The subpopulation  $S|_G$  uses half of the individuals from  $P|_G$  and half of the individuals from  $A|_G$ . Regarding the GPP index, values  $\alpha_k = \frac{k}{10}$ ,  $k > 0$  and  $\delta_k = (m + 1) \cdot (\alpha_k + \delta_{k-1})$ ,  $k > 1$  are used (in accordance with Table 1).

### 4.1. The 3-bar truss design problem

The first problem is the well-known bi-objective 3-bar truss design. The truss design problem is a classical MOP benchmark statement to test algorithms, as well as decision making step procedures. The truss parameters proposed in [48] are used. Two objectives are minimized: deflection squared ( $J_1(\boldsymbol{\theta})$ , [cm<sup>2</sup>]) and total volume ( $J_2(\boldsymbol{\theta})$ , [cm<sup>3</sup>]), where each bar is constrained to a maximum stress  $\sigma < 2E + 008$  [Pa]. Characteristics to be evaluated with this benchmark problem are:

- The implicit constraint handling mechanism with  $J_{gpp}(\boldsymbol{\varphi})$  and the pruning technique.
- The capacity to build a  $\mathbf{T}\text{-}\mathbf{J}_P^*$ ,  $\mathbf{D}\text{-}\mathbf{J}_P^*$  and  $\mathbf{HD}\text{-}\mathbf{J}_P^*$  in one single run.
- Pareto front approximation control size.

As a rule of thumb, it will be stated that the DM looks for a Pareto set approximation with  $10 \cdot m = 20$  solutions (approximately). Preferences stated for this problem are shown in



Table 1: Guidelines for spMODE-II's parameters tuning.

Parameter	Value	Comments
DE algorithm		
$F$ (Scaling factor)	0.5	Recognized as good initial choice according to [52].
$Cr$ (Crossover rate)	[0.8, 1.0]	Values recognized for non-separable problems according to [63, 54].
	[0.1, 0.2]	Values recognized for separable problems according to [63, 54].
	0.5	Trade-off value for separable and non-separable problems. Default value used (for example) by MOEA/D algorithm [10].
$N_p$ (Population)	50	While a <i>five to ten times the number of decision variables</i> rule has been recognized as a rule of thumb [52] for single objective optimisation, here a default size of 50 individuals is proposed.
Pruning mechanism		
$\beta_\epsilon$ (Arcs)	$10 \cdot \overbrace{[m, \dots, m]}^{m-1}$	It is proposed for $m$ -objective problems, to bound the grid size to $m^{m-1}$ hyper-spherical sectors.
Pertinency mechanism		
$J_{gpp}^{max}$	$J_{gpp}(\mathbf{J}^T)$	Proposed as default value. Only solutions with their $m$ objectives in the tolerable region might appear in $\mathbf{J}_P^*$ . It will be modified by the algorithm if the size control mechanism (Algorithm 7) is activated.
$\overline{car}(\mathbf{J}_P^*)$	$10 \cdot m$	Proposed as default value. In accordance with [31].
$\alpha_k$	$\frac{k}{10}, k > 0$	Proposed as default value in accordance with Equation (19).
$\delta_k$	$(m + 1) \cdot (\alpha_k + \delta_{k-1}), k > 1$	Proposed as default value in accordance with Equation (23).

Table 2: Preferences for the 3-bar truss design problem. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set A											
Objective	$\leftarrow$ $J_i^0$	HD	$\rightarrow \leftarrow$ $J_i^1$	D	$\rightarrow \leftarrow$ $J_i^2$	T	$\rightarrow \leftarrow$ $J_i^3$	U	$\rightarrow \leftarrow$ $J_i^4$	HU	$\rightarrow$ $J_i^5$
$J_1(\boldsymbol{\theta})$ [cm <sup>2</sup> ]	0.00		0.06		0.08		0.10		0.20		0.50
$J_2(\boldsymbol{\theta})$ [cm <sup>3</sup> ]	0.00		550		650		750		1000		1500
$G_1(\boldsymbol{\theta})$ [Pa]	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008
$G_2(\boldsymbol{\theta})$ [Pa]	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008
$G_3(\boldsymbol{\theta})$ [Pa]	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008

Table 2. Notice that constraints are incorporated as additional objectives; since the designer is just interested in calculating solutions with  $G_i(\boldsymbol{\theta}) < 2E+008$  [Pa], no further specifications are required below the tolerable value. These additional objectives are used to calculate the  $J_{gpp}(\boldsymbol{\varphi})$  index, but they will not be used for partitioning the space, unless required.

The results of one single typical run are shown in Figure 11. The following observations can be made:

1. In Figure 11a, a typical run with a grid size of 500 spherical sectors is shown. In this case, it is important to notice that the algorithm is capable of reaching the **HD- $J_P^*$** , but using computational resources without regarding the areas of interest to the DM.
2. In Figure 11b, the algorithm is executed again, with the same parameters, but using the  $J_{gpp}(\boldsymbol{\varphi})$  index. In this case, the algorithm spends resources taking care of **T-HypV**.
3. Finally, in Figure 11c, the algorithm is executed again, with the size control mechanism. Note how the algorithm concentrates on the **HD-HypV**, sacrificing all solutions in the **T-HypV**, and retaining few solutions from the desirable Pareto front.

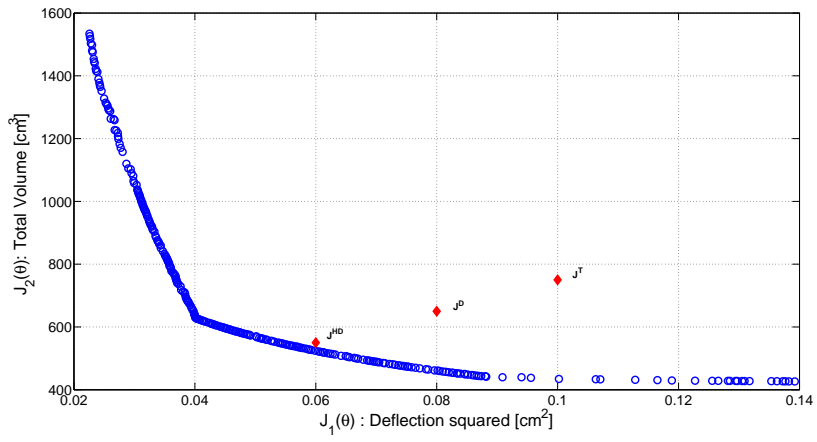
Certainly, for bi-objective problems (most of the cases) it is not difficult to attain a Pareto front approximation such as the one in Figure 11a and apply successive filtering to reach the highly desirable region. Nevertheless, with cost function evaluations with high computational loads (complex simulations for example) it could be a desirable characteristic to evolve quickly to the desirable zone.

Usually, it is not difficult to perform an analysis on bi-objective problems, for trade-off and preference articulation. To evaluate the flexibility of the  $J_{gpp}(\boldsymbol{\varphi})$  index a problem with three objectives is analysed below.

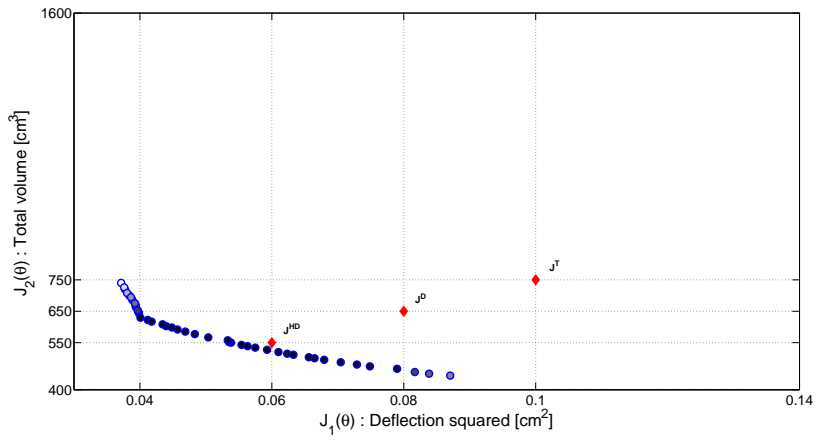
#### 4.2. The DTLZ2 benchmark problem

The second benchmark example is the DTLZ2 problem [64], with three objectives and ten decision variables. The Pareto front is the surface contained on the first quadrant of a hypersphere with unitary radius. It is used to show the following characteristics:

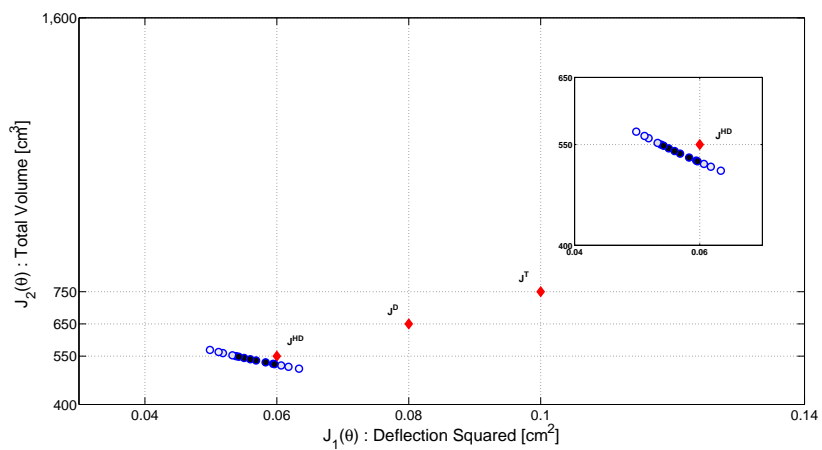
- Capacity to build a **T- $J_P^*$**



(a) No preferences used.



(b) GPP is used. Solutions on the tolerable hypervolume are only considered.



(c) Size control. Original grid size is always preserved

Figure 11: Algorithm performance in the bi-objective 3 bar truss design benchmark problem.

Table 3: Preferences for the DTLZ2 benchmark problem. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

		Preference Set A									
Objective	$\leftarrow$	HD	$\rightarrow \leftarrow$	D	$\rightarrow \leftarrow$	T	$\rightarrow \leftarrow$	U	$\rightarrow \leftarrow$	HU	$\rightarrow$
	$J_i^0$		$J_i^1$		$J_i^2$		$J_i^3$		$J_i^4$		$J_i^5$
$J_1(\boldsymbol{\theta})$ [-]	0.00		0.05		0.10		0.40		1.00		10
$J_2(\boldsymbol{\theta})$ [-]	0.00		0.30		0.40		0.60		1.00		10
$J_3(\boldsymbol{\theta})$ [-]	0.00		0.50		0.80		0.90		1.00		10

- Capacity to build a  $\mathbf{T}\text{-}\mathbf{J}_P^*$ , with solutions with at least one objective in the desirable hypervolume (or equivalently, solutions with two objectives in the tolerable value).

Preferences stated for this problem are shown in Table 3. Furthermore, it will be stated that the DM has some interest in keeping the objectives in the desirable zone and is willing to accept two of them (and not more) in the tolerable region; for this reason,  $J_{gpp}^{max}$  is adapted to  $J_{gpp}^{max} = J_{gpp}([J_1^3, J_2^3, J_3^2])$ .

In Figure 12, results for the following algorithm executions are shown:

- Execution with no preferences (\*), used for comparison purposes.
- Execution to build a  $\mathbf{T}\text{-}\mathbf{J}_P^*$  (blue circles  $\circ$ ).
- Execution to build a  $\mathbf{T}\text{-}\mathbf{J}_P^*$  with at most two objectives in the tolerable zone (black diamonds  $\diamond$ ).

Regarding the  $\mathbf{T}\text{-}\mathbf{J}_P^*$ , the algorithm reaches the zone according to the preferences defined earlier. The (apparently) irregular distribution is owing to the selection of the most preferable solution inside each spherical sector. With regard to the tolerable Pareto front with at most two objectives in the tolerable region, the algorithm is capable of avoiding solutions in the middle of the tolerable region ( $J_2(\boldsymbol{\theta}) \approx [0.4, 0.5]$ ), which corresponds to solutions with  $J_{gpp}(\varphi) > J_{gpp}([0.4, 0.6, 0.8])$  and consequently, uninteresting to the DM (any solution with  $J_{gpp}([0.4, 0.6, 0.8]) < J_{gpp}(\varphi) \leq J_{gpp}([0.4, 0.6, 0.9])$  has its three objectives in the tolerable region). This fact shows the flexibility of the approach of evolving its population towards different regions of interest and avoiding regions that are uninteresting to the DM.

The flexibility in reaching regions of interest and the capacity to adapt the archive size according to the desired number of solutions, enables the algorithm to deal with many-objective optimisation. A many-objective problem is then used to show the ability of the approach to deal with higher dimensions.

### 4.3. Parametric controller tuning

The following example is a parametric controller design for the control benchmark proposed at the American Control Conference (ACC) [65]. The MOP statement described in

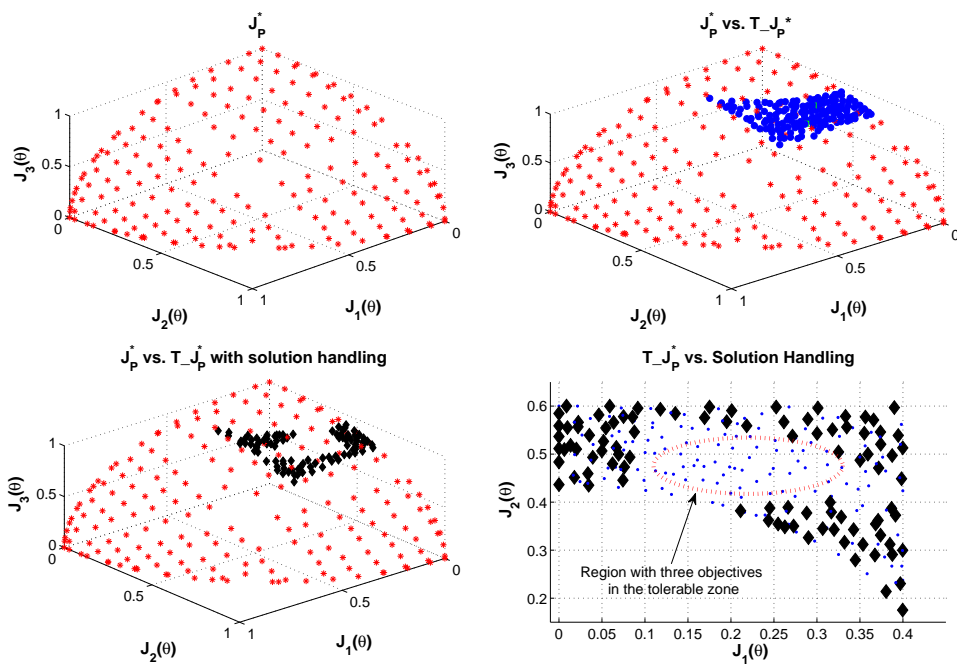


Figure 12: Performance of the tolerable solution handling mechanism. Pareto front approximation (red- $*$ ), tolerable Pareto front (blue-o) and Pareto front approximation with at most two values in the tolerable zone (black- $\diamond$ ) are depicted.

Table 4: Preferences for parametric controller tuning example. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Objective	Preference Set A										
	← $J_i^0$	HD	→ ← $J_i^1$	D	→ ← $J_i^2$	T	→ ← $J_i^3$	U	→ ← $J_i^4$	HU	→ $J_i^5$
$J_1(\boldsymbol{\theta})$ [-]	-0.01		-0.005		-0.001		-0.0005		-0.0001		0
$J_2(\boldsymbol{\theta})$ [-]	0.85		0.900		1.000		1.5000		2.0000		10
$J_3(\boldsymbol{\theta})$ [sec]	14.00		16.000		18.000		21.0000		25.0000		50
$J_4(\boldsymbol{\theta})$ [-]	0.50		0.900		1.200		1.4000		1.5000		10
$J_5(\boldsymbol{\theta})$ [-]	0.50		0.700		1.000		1.5000		2.0000		10
$J_6(\boldsymbol{\theta})$ [sec]	10.00		11.000		12.000		14.0000		15.0000		50
$G_1(\boldsymbol{\theta})$ [-]	2.00		2.000		2.000		2.0000		5.0000		10

[27] is used. It has six objectives: robust stability ( $J_1(\boldsymbol{\theta})$  [-]); maximum control effort ( $J_2(\boldsymbol{\theta})$  [-]); worst case settling time ( $J_3(\boldsymbol{\theta})$  [sec]); noise sensitivity ( $J_4(\boldsymbol{\theta})$  [-]); nominal control effort ( $J_5(\boldsymbol{\theta})$  [-]); and nominal settling time ( $J_6(\boldsymbol{\theta})$  [sec]). Additionally, a constraint on the overshoot in the nominal model is imposed:  $G_1(\boldsymbol{\theta})[-] < 2$ . One controller structure will be evaluated:

$$C(s) = \frac{\theta_1 s^2 + \theta_2 s + \theta_3}{s^3 + \theta_4 s^2 + \theta_5 s + \theta_6} \quad (26)$$

Characteristics to be evaluated are:

- The implicit constraint handling mechanism with GPP and the pruning technique.
- Capacity to build  $\mathbf{T}\text{-}\mathbf{J}_P^*$ ,  $\mathbf{D}\text{-}\mathbf{J}_P^*$  and  $\mathbf{HD}\text{-}\mathbf{J}_P^*$  in a many-objective optimisation framework.
- Capacity for Pareto front approximation control size.

The preference set in Table 4 is used. The calculated Pareto front for reference is depicted in Figure 13 using the LD-Tool<sup>11</sup> which is an application developed in Matlab© for Pareto front visualisation using level diagrams [27]. With the LD-Tool, a color scheme can be used to depict interesting solutions for the DM according to her or his preference set definition. A geometrical remark is relevant in the figure: in  $J_1(\boldsymbol{\theta})$  two different and isolated regions in the Pareto front fulfil the designer's preferences.

The results from Figure 13 can be compared with that of the ev-MOGA algorithm presented in [27] and provided within the LD-Toolbox, where the preference set is used  $a$

<sup>11</sup>GUI for Matlab is available at: <http://www.mathworks.com/matlabcentral/fileexchange/24042>

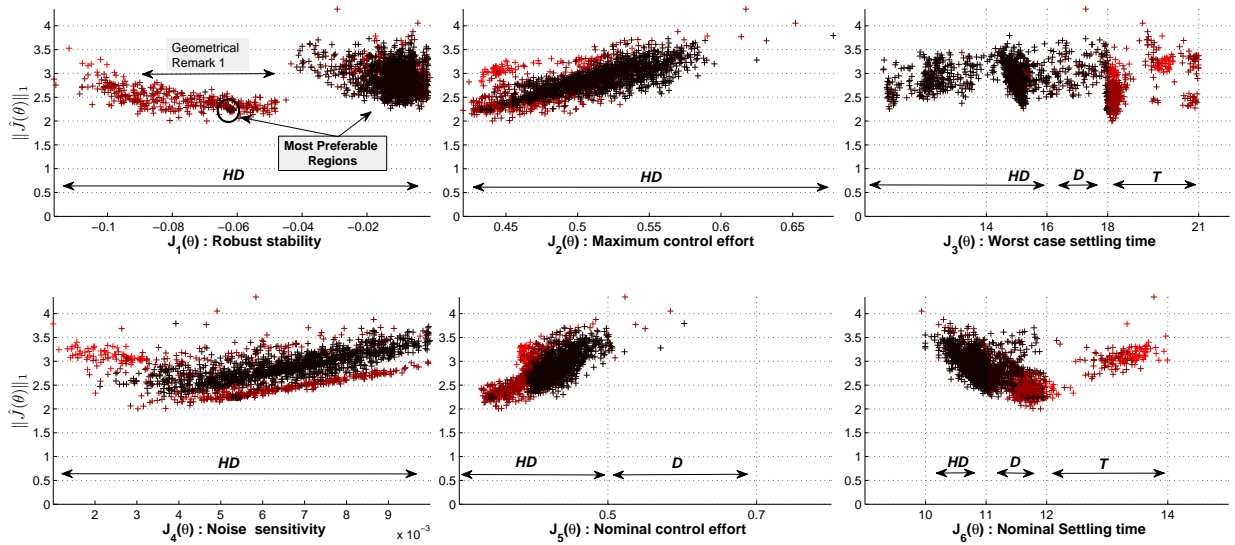


Figure 13: Visualization of the preferable Pareto front of the parametric controller tuning example using level diagrams.

*posteriori*. While it is possible to identify the pertinent regions in the Pareto front, many computational resources were used in the remaining areas of the objective space. With the usage of GPP for preference articulation it is possible to focus the searching process in the area of interest. As a consequence in this case, the spMODE-II is capable of finding solutions in the **T\_HypV** but the ev-MOGA was not. This is due to the additional information embedded in the optimisation process provided by the DM.

#### 4.4. Performance evaluation with other approaches

The last example is the pollution monitoring MOP stated and solved by the IBEA algorithm in [42]. The IBEA algorithm is an indicator based MOEA [66], which uses the  $\epsilon$  indicator [67] to evolve the entire Pareto front approximation. The MOP states five objectives, each representing the expected information loss in five different monitoring stations.

$$\min \mathbf{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta}), J_3(\boldsymbol{\theta}), J_4(\boldsymbol{\theta}), J_5(\boldsymbol{\theta})] \quad (27)$$

$$J_1(\boldsymbol{\theta}) = -u_1(\theta_1, \theta_2) - u_2(\theta_1, \theta_2) - u_3(\theta_1, \theta_2) + 10 \quad (28)$$

$$J_2(\boldsymbol{\theta}) = -u_1(\theta_1 - 1.2, \theta_2 - 1.5) - u_2(\theta_1 - 1.2, \theta_2 - 1.5) - u_3(\theta_1 - 1.2, \theta_2 - 1.5) + 10 \quad (29)$$

$$J_3(\boldsymbol{\theta}) = -u_1(\theta_1 + 0.3, \theta_2 - 3.0) - u_2(\theta_1 + 0.3, \theta_2 - 3.0) - u_3(\theta_1 + 0.3, \theta_2 - 3.0) + 10 \quad (30)$$

$$J_4(\boldsymbol{\theta}) = -u_1(\theta_1 - 1.0, \theta_2 + 0.5) - u_2(\theta_1 - 1.0, \theta_2 + 0.5) - u_3(\theta_1 - 1.0, \theta_2 + 0.5) + 10 \quad (31)$$

$$J_5(\boldsymbol{\theta}) = -u_1(\theta_1 - 0.5, \theta_2 - 1.7) - u_2(\theta_1 - 0.5, \theta_2 - 1.7) - u_3(\theta_1 - 0.5, \theta_2 - 1.7) + 10 \quad (32)$$

where:

$$u_1(\theta_1, \theta_2) = 3(1 - \theta_1)^2 \exp^{-\theta_1^2 - (\theta_2 + 1)^2} \quad (33)$$

$$u_2(\theta_1, \theta_2) = -10(\theta_1/4 - \theta_1^3 - \theta_2^5) \exp^{-\theta_1^2 - \theta_2^2} \quad (34)$$

$$u_3(\theta_1, \theta_2) = \frac{1}{3} \exp^{-(\theta_1 + 1)^2 - \theta_2^2} \quad (35)$$

subject to:

$$\theta_1 \in [-4.9, 3.2] \quad (36)$$

$$\theta_2 \in [-3.5, 6.0] \quad (37)$$

Characteristics to be evaluated are:

- The performance of the strategy with respect to state of the art techniques.
- The capacity to build multiple  $\mathbf{T}\text{-}\mathbf{J}_P^*$ ,  $\mathbf{D}\text{-}\mathbf{J}_P^*$  and  $\mathbf{HD}\text{-}\mathbf{J}_P^*$  in many-objective optimisation instances with different preference conditions. This will evaluate indirectly the capacity to deal with *subset against subset* priorities.
- The capacity for Pareto front approximation size control.

Two instances will be evaluated: a single physical matrix A and two simultaneous physical matrices B, and C (Table 5). In each case, 2000 function evaluations are used and 50 solutions are required by the DM in  $\mathbf{J}_P^*$ .

The proposal's performance will be evaluated and compared against three different techniques:



- An IBEA [42] using DE algorithm ( $\rho = 0.05$ ), hereafter denoted as IB-MODE. This strategy is selected because it is a *state of the art* technique for handling simultaneous preference conditions.
- A spMODE algorithm using the reference point based multi-objective optimization technique described in [47], hereafter denoted as RP-spMODE. This technique is selected because it is consistent with the spMODE algorithm structure, and it could be used as a *customized* norm in its pruning mechanism.
- A pure stochastic sampling approach, using the same function evaluations budget. This is used as a base test to evaluate the usability of the approaches above. This is because it has been remarked that sampling procedures could be more effective in many-objective optimization statements [43].

In the cases of IB-MODE and RP-spMODE, if they use the **T\_Vector**, **D\_Vector** or **HD\_Vector**, they will evolve the Pareto front approximation towards the **T\_HypV**, **D\_HypV** and **HD\_HypV**. In all cases, the comparison is performed with *a priori* techniques within the classification *dominance is essential* to appreciate their differences when *objective value against objective value* is important for the designer. *Progressive* techniques are intentionally omitted, since it is not possible to perform a fair comparison, given the structural differences regarding the input of the DM's preferences. In the same way, techniques within the classification *objective against objective* are not included because they deal with a different knowledge of the DM's preferences.

Parameters for spMODE-II, IB-MODE and RP-spMODE algorithms are stated (accordingly for each case) as depicted in Table 1 for non-separable problems ( $F = 0.5$  and  $CR = 0.9$ ). A total of 201 independent runs are carried out for all cases.

In Tables 6 and 7 the results of 201 independent runs for the best, median, worst and mean values of the achieved **HD\_HypV**, **D\_HypV** and **T\_HypV** are depicted. Statistical differences have been validated with the Wilcoxon test and Bonferroni correction at 95% significance level [68]. Regarding instance 1, none of the algorithms are capable of reaching the **HD\_HypV**; it is also evident the difference in performance between spMODE-II/IB-MODE against RP-spMODE/sampling approaches. The spMODE-II algorithm is able to achieve a better performance than IB-MODE for **D\_HypV**, at the cost of degrading the **T\_HypV**.

Regarding instance 2, only the spMODE-II algorithm is capable of finding solutions in the **HD\_HypV** for both preference sets simultaneously. Again, spMODE-II presents the same behavior when compared with IB-MODE for **D\_HypV** and **T\_HypV**. This is due to the implicit pressure on the  $J_{gpp}(\theta)$  index to continue improving the obtained solutions from the **T\_HypV** to the **HD\_HypV**.

Pair comparisons for 50% attainment surfaces are depicted in Figures 15 and 16 as described in [28] for MOEAs comparison in many-objective optimisation instances by means of level diagrams.<sup>12</sup> Taxonomy to identify the visualisations is adopted from the same ref-

---

<sup>12</sup>Available for Matlab at: <http://www.mathworks.com/matlabcentral/fileexchange/39458>

Table 5: Preference Set A for Example 4. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set A											
Objective	$\leftarrow$	HD	$\rightarrow \leftarrow$	D	$\rightarrow \leftarrow$	T	$\rightarrow \leftarrow$	U	$\rightarrow \leftarrow$	HU	$\rightarrow$
	$J_i^0$		$J_i^1$		$J_i^2$		$J_i^3$		$J_i^4$		$J_i^5$
$J_1(\boldsymbol{\theta})$ [-]	6		7		9		10		11		12
$J_2(\boldsymbol{\theta})$ [-]	6		7		8		10		11		12
$J_3(\boldsymbol{\theta})$ [-]	6		7		9		10		11		12
$J_4(\boldsymbol{\theta})$ [-]	6		7		8		10		11		12
$J_5(\boldsymbol{\theta})$ [-]	6		7		9		10		11		12
Preference Set B											
Objective	$\leftarrow$	HD	$\rightarrow \leftarrow$	D	$\rightarrow \leftarrow$	T	$\rightarrow \leftarrow$	U	$\rightarrow \leftarrow$	HU	$\rightarrow$
	$J_i^0$		$J_i^1$		$J_i^2$		$J_i^3$		$J_i^4$		$J_i^5$
$J_1(\boldsymbol{\theta})$ [-]	5		10		11		12		13		15
$J_2(\boldsymbol{\theta})$ [-]	5		9		10		11		12		15
$J_3(\boldsymbol{\theta})$ [-]	5		8		9		10		11		15
$J_4(\boldsymbol{\theta})$ [-]	5		7		8		9		10		15
$J_5(\boldsymbol{\theta})$ [-]	5		6		7		8		9		15
Preference Set C											
Objective	$\leftarrow$	HD	$\rightarrow \leftarrow$	D	$\rightarrow \leftarrow$	T	$\rightarrow \leftarrow$	U	$\rightarrow \leftarrow$	HU	$\rightarrow$
	$J_i^0$		$J_i^1$		$J_i^2$		$J_i^3$		$J_i^4$		$J_i^5$
$J_1(\boldsymbol{\theta})$ [-]	5		8		9		10		14		15
$J_2(\boldsymbol{\theta})$ [-]	5		8		9		10		14		15
$J_3(\boldsymbol{\theta})$ [-]	5		11		12		13		14		15
$J_4(\boldsymbol{\theta})$ [-]	5		11		12		13		14		15
$J_5(\boldsymbol{\theta})$ [-]	5		11		12		13		14		15

Table 6: Hypervolume achieved in 201 runs for a single physical matrix. Statistical differences (if any) according to the Wilcoxon test among spMODE-II ( $\checkmark$ ), IB-MODE ( $\dagger$ ), RP-spMODE ( $\ddagger$ ), and the Sampling approach ( $\S$ ) are depicted for each instance.

		Single Physical Matrix			
		Preference Set A			
		spMODE-II $\checkmark$	IB-MODE $\dagger$	RP-MODE $\ddagger$	Sampling $\S$
HD	Best	0.0000	0.0000	0.0000	0.0000
	Median	0.0000	0.0000	0.0000	0.0000
	Worst	0.0000	0.0000	0.0000	0.0000
	Mean	0.0000	0.0000	0.0000	0.0000
	std	0.0000	0.0000	0.0000	0.0000
D	Best	<b>1.2533</b>	0.6902	0.1291	0.3610
	Median	<b>1.1586</b> $\dagger, \ddagger, \S$	0.6342 $\checkmark, \ddagger, \S$	0.0000 $\checkmark, \dagger$	0.0000 $\checkmark, \dagger$
	Worst	<b>0.5880</b>	0.5611	0.0000	0.0000
	Mean	<b>1.1320</b>	0.6324	0.0060	0.0197
	std	0.0971	<b>0.0219</b>	0.0183	0.0690
T	Best	<b>260.0377</b>	259.1223	93.2508	193.5246
	Median	146.6786 $\dagger, \ddagger, \S$	<b>242.2504</b> $\checkmark, \ddagger, \S$	28.1823 $\checkmark, \dagger, \S$	73.2174 $\checkmark, \dagger, \ddagger$
	Worst	123.8105	<b>216.6591</b>	5.3231	1.5615
	Mean	184.3619	<b>242.0966</b>	28.0505	75.0959
	std	47.7800	<b>7.2756</b>	12.7703	45.1002

Table 7: Hypervolume achieved in 201 runs for simultaneous physical matrixes. Statistical differences (if any) according to the Wilcoxon test among spMODE-II ( $\checkmark$ ), IB-MODE ( $\dagger$ ), RP-spMODE ( $\ddagger$ ) and the Sampling approach ( $\S$ ) are depicted for each instance.

		Simultaneous Physical Matrixes							
		Preference B				Preference C			
		spMODE-II	IB-MODE	RP-spMODE	Sampling	spMODE-II	IB-MODE	RP-spMODE	Sampling
HD	Best	<b>0.1782</b>	0.0148	0.0000	0.0331	<b>0.3987</b>	0.0000	0.2055	0.1256
	Median	<b>0.1241</b> $\dagger, \ddagger, \S$	0.0000 $\checkmark$	0.0000 $\checkmark$	0.0000 $\checkmark$	<b>0.3147</b> $\dagger, \ddagger, \S$	0.0000 $\checkmark, \dagger$	0.1331 $\checkmark, \dagger, \S$	0.0000 $\checkmark, \dagger$
	Worst	<b>0.0293</b>	0.0000	0.0000	0.0000	<b>0.0952</b>	0.0000	0.0002	0.0000
	Mean	<b>0.1187</b>	0.00007	0.0000	0.0003	<b>0.3091</b>	0.0000	0.1279	0.0084
	std	<b>0.0308</b>	0.0010	0.0000	0.0027	<b>0.0550</b>	0.0000	0.0400	0.0233
D	Best	<b>56.8633</b>	52.2928	13.3139	29.9494	<b>123.9829</b>	18.5836	60.0368	115.6656
	Median	<b>49.7769</b> $\dagger, \ddagger, \S$	9.1762 $\checkmark, \ddagger, \S$	0.7073 $\checkmark, \dagger$	0.0000 $\checkmark, \dagger$	<b>47.7898</b> $\dagger, \ddagger, \S$	6.0634 $\checkmark, \ddagger, \S$	35.7351 $\checkmark, \dagger$	31.8531 $\checkmark, \dagger$
	Worst	<b>21.0424</b>	0.0000	0.0000	0.0000	<b>20.8970</b>	0.0000	19.6379	0.0000
	Mean	<b>48.1316</b>	8.7044	1.3396	5.6535	<b>47.7064</b>	5.8899	36.0081	34.6440
	std	<b>6.2359</b>	6.6696	1.9116	7.8667	9.9826	<b>4.6920</b>	8.6873	24.3320
T	Best	515.4334	<b>585.4359</b>	193.5993	404.7116	<b>1.9292E+003</b>	<b>1.7819E+003</b>	1.2116E+003	1.8011E+003
	Median	429.8147 $\dagger, \ddagger, \S$	<b>461.2593</b> $\checkmark, \ddagger, \S$	67.4729 $\checkmark, \dagger, \S$	125.5715 $\checkmark, \dagger, \ddagger$	1.3083E+003 $\dagger, \ddagger, \S$	<b>1.5312E+003</b> $\checkmark, \ddagger, \S$	0.7290E+003 $\checkmark, \dagger, \S$	0.9802E+003 $\checkmark, \dagger, \ddagger$
	Worst	218.4111	<b>257.7210</b>	9.6223	0.0000	<b>1.0425E+003</b>	0.8677E+003	0.2501E+003	0.1507E+003
	Mean	423.7853	<b>456.3398</b>	69.6058	141.8065	1.3138E+003	<b>1.5044E+003</b>	0.7350E+003	0.9970E+003
	std	51.6358	<b>40.5184</b>	27.7626	112.6920	0.1173E+003	<b>0.1073E+003</b>	0.1349E+003	0.3288E+003

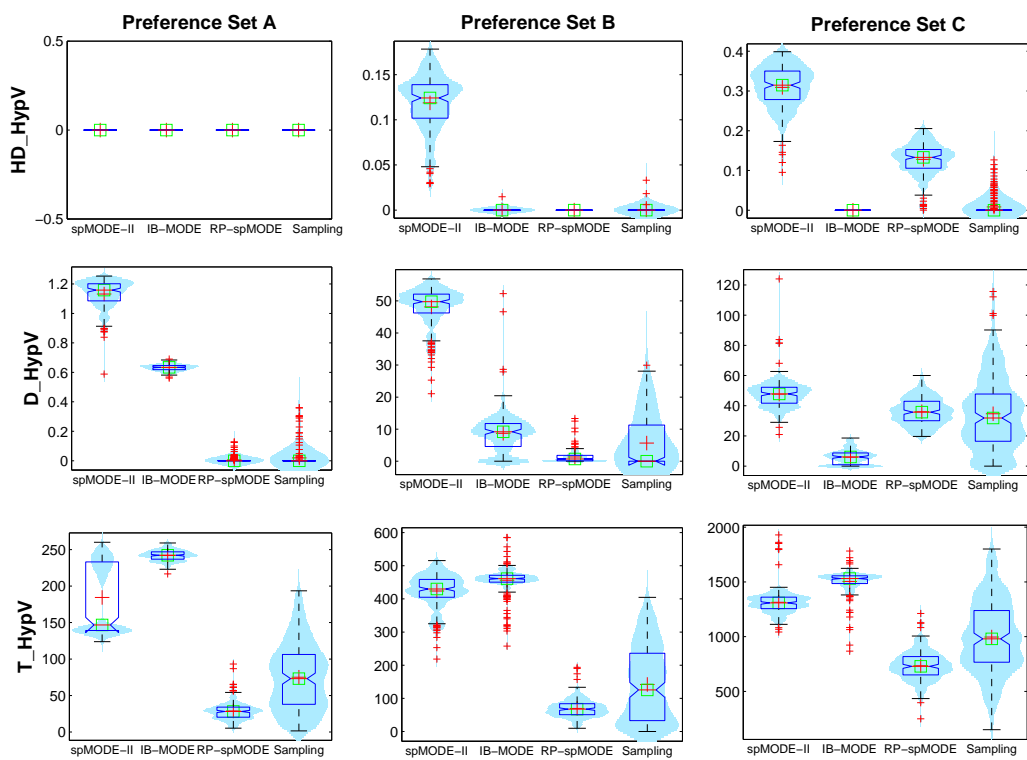


Figure 14: Hypervolume achieved in 201 runs. Boxplot diagrams and density estimations are shown in each case.

erence<sup>13</sup>. In this visualization, any surface above 1.0 is dominated, while any surface below 1.0 dominates the other.

It is possible to notice the following remarks on Figure 15:

**Remark 1:** In instance 1 (single preference condition) the 50% attainment surface of the spMODE-II algorithm fails to converge at the **D\_HypV**. This impossibility is due to Objective 1: the algorithm fails to handle this objective to reach the **T\_J<sub>p</sub>\***. As a consequence, the IB-MODE algorithm dominates this attainment surface portion.

**Remark 2:** In the same case, when the 50% attainment surface of the spMODE-II algorithm converges at the **D\_HypV**, it tends to dominate the IB-MODE algorithm.

**Remark 3:** According to instance 2 (simultaneous preference conditions), there is a change in the covering for  $J_1(\boldsymbol{\theta})$  and  $J_5(\boldsymbol{\theta})$  (Figure 15b). That could indicate structural differences between algorithms, since one of them dominates the other in one of the objectives, at the price of being dominated in the other.

**Remark 4:** According to Table 7, the IB-MODE is capable of reaching the **HD\_HypV** of the preference set B, but incapable of reaching the **HD\_HypV** of the preference set C. In Figure 15b, it is possible to appreciate this fact and it is possible to specify where it happens: in Objective  $J_1(\boldsymbol{\theta})$  the IB-MODE fails to reach the **HD\_HypV** of preference set C.

Regarding Figure 16, the following remarks can be made:

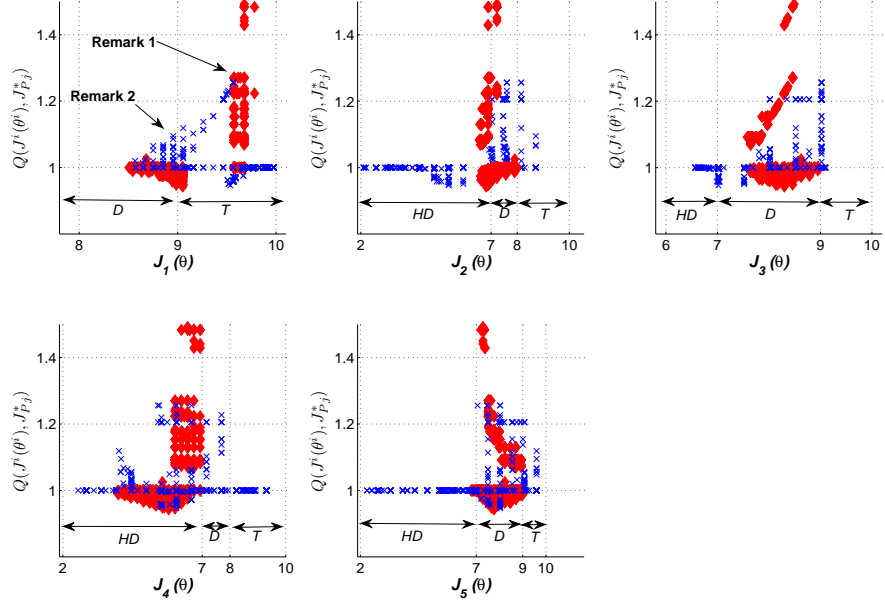
**Remark 5:** In instance 1 (single preference condition) the 50% attainment surface of the spMODE-II algorithm consistently dominates the RP-spMODE algorithm.

**Remark 6:** According to Table 7, the RP-spMODE is capable of reaching the **HD\_HypV** of the preference set C, but incapable of reaching the **HD\_HypV** of the preference set B. In Figure 16b, it is possible to appreciate this fact and it is possible to specify where it happens: in Objective  $J_3(\boldsymbol{\theta})$  the RP-spMODE fails to reach the **HD\_HypV** of preference set B.

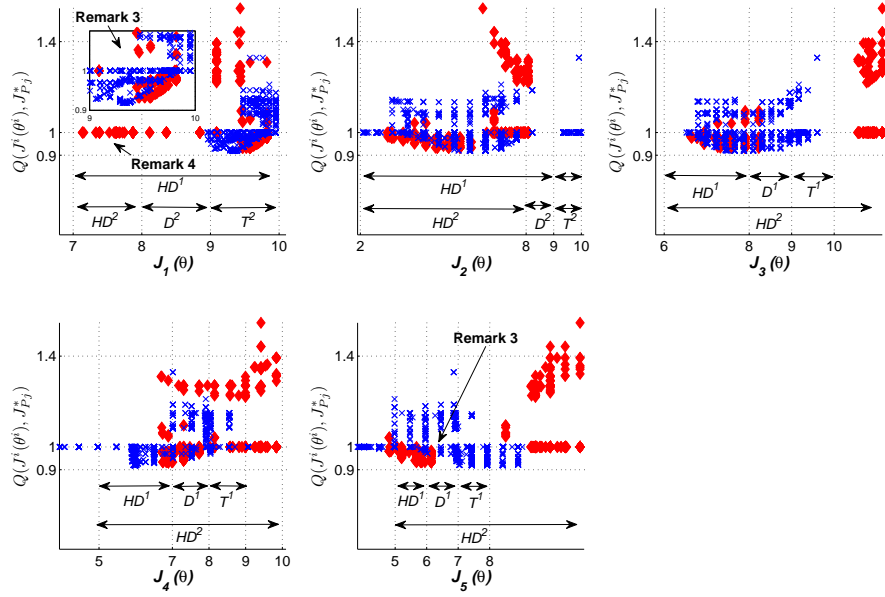
In accordance with the above, the spMODE-II algorithm shows its efficacy in handling simultaneous preference conditions to approximate pertinent Pareto front approximations for the DM.

---

<sup>13</sup>*LD/front/measure*. For example,  $LD/\mathbf{J}_p^*/\|\hat{\mathbf{J}}(\boldsymbol{\theta})\|_2$ , means that a visual representation of Pareto front approximation  $\mathbf{J}_p^*$  with 2-norm in LD is presented.

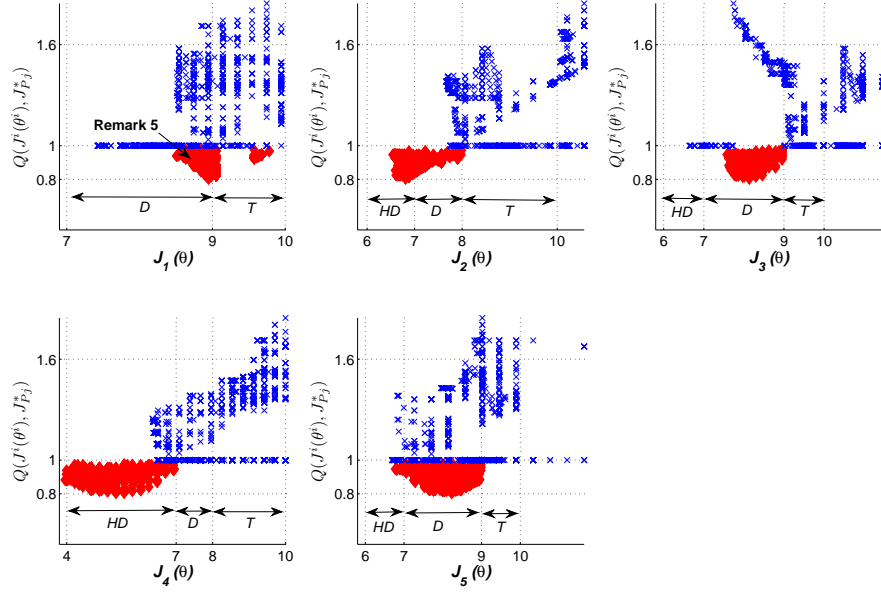


(a) Single preference condition

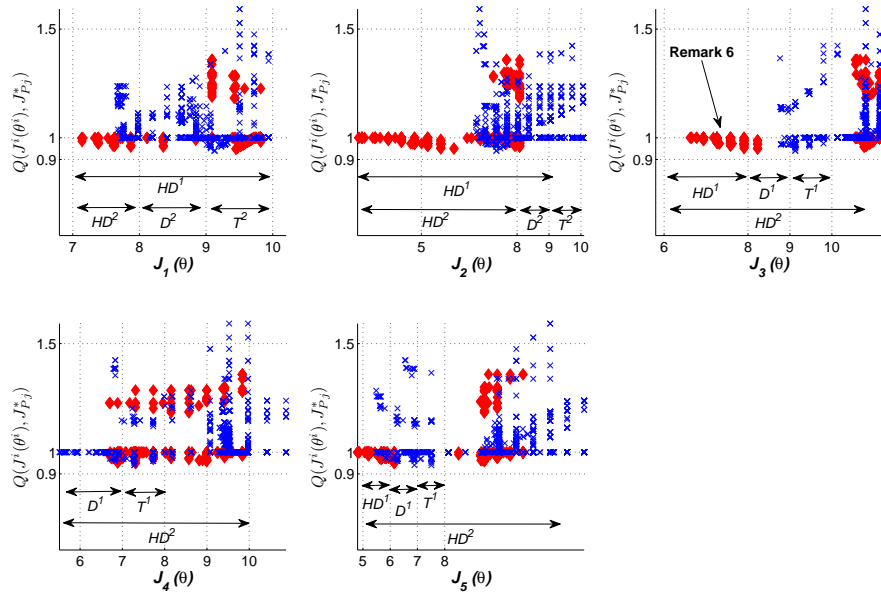


(b) Simultaneous preference conditions

Figure 15:  $LD / \left\{ \widetilde{J}_{p1}^*{}^{50\%}, \widetilde{J}_{p2}^*{}^{50\%} \right\} / Q(J^i(\theta^i), J_{p_j}^*)$  visualisation for attainment surface performance. Red  $\diamond$  for spMODE-II and blue  $x$  for IB-MODE are used. Detailed explanation on the remarks is given within the paper.



(a) Single preference condition



(b) Simultaneous preference conditions

Figure 16:  $LD / \left\{ \widetilde{J}_{p1}^{*50\%}, \widetilde{J}_{p2}^{*50\%} \right\} / Q(J^i(\theta^i), J_{p_j}^*)$  visualisation for attainment surface performance. Red  $\diamond$  for spMODE-II and blue  $x$  for RP-spMODE are used. Detailed explanation on the remarks is given within the paper.

## 5. Conclusions

In this work, a mechanism for preference handling in EMO using GPP has been presented. Techniques for preference handling could be desirable characteristics for an MOEA, since they improve the pertinency of solutions. Therefore, this mechanism could facilitate the DM's task of analyzing and selecting a preferable solution from the approximated Pareto front. The embedment of the articulation of preferences in an MOEA could reinforce the holistic MOOD procedure.

The developed approach fulfills the requirement for preference coding in an intuitive and meaningful scheme in order to generate pertinent solutions. As a consequence, pertinent approximations of the Pareto front are built. Furthermore, the approach has shown to be effective for constrained and many-objective optimisation statements.

According to the examples provided, a competitive algorithm has been developed. Although it has been presented as a specific algorithm, the technique could be incorporated in other EMO approaches. The developed algorithm (spMODE-II) is flexible for articulating preferences and is DM-oriented since it provides features for multiple preference conditions, controls the size of the approximated front (based on preferences), and is able to handle the level of tolerance of a design alternative.

The advantages of using this framework are:

1. DM preferences are coded in an intuitive and meaningful language.
2. A constraint handling mechanism is included in the optimisation statement.
3. The pruning mechanism can be refined by specifying which solutions are not allowed to remain in the Pareto front approximation, according to the DM's preferences.
4. A control mechanism can be created for the quantity of solutions required in the Pareto front approximation, with regard to the solutions that interest the designer.
5. The framework enables the algorithm to define priorities on design alternatives according to an *Objective value against objective value* basis (see Algorithm 6).
6. The algorithm can handle multiple preference conditions simultaneously and approximate Pareto optimal solutions for each – independently of the Pareto front shape.

However, its limitations are:

1. The proposal assumes that the designer has an idea about the preference values. This may not be the case, and sometimes for the designer it is only important to prioritize some of the objectives (*objective against objective*). In such cases, other mechanisms should be used.
2. This approach states the preference conditions only in an *a priori* step; therefore, it represents a limitation when compared with *progressive* approaches. Further research will focus on this topic.

## Acknowledgment

This work was partially supported by the FPI-2010/19 grant and the PAID-2011/2732 project from the Universitat Politècnica de València and the projects TIN2011-28082, ENE2011-25900 from the Spanish Ministry of Economy and Competitiveness.



- [1] A. Messac and C. Mattson, “Generating well-distributed sets of Pareto points for engineering design using Physical Programming,” *Optimization and Engineering*, vol. 3, pp. 431–450, 2002.
- [2] K. M. Miettinen, *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1998.
- [3] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, “Multi-criteria decision making,” in *Evolutionary Algorithms for Solving Multi-Objective Problems*, ser. Genetic and Evolutionary Computation. Springer US, 2007, pp. 515–545.
- [4] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. M. Herrero, “Multiobjective evolutionary algorithms for multivariable PI controller tuning,” *Expert Systems with Applications*, vol. 39, pp. 7895 – 7907, 2012.
- [5] I. Das and J. Dennis, “Normal-boundary intersection: a new method for generating the Pareto surface in non-linear multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, pp. 631 – 657, 1998.
- [6] A. Messac, A. Ismail-Yahaya, and C. Mattson, “The normalized normal constraint method for generating the Pareto frontier,” *Structural and Multidisciplinary Optimization*, no. 25, pp. 86 – 98, 2003.
- [7] J. Sanchis, M. Martínez, X. Blasco, and J. V. Salcedo, “A new perspective on multiobjective optimization by enhanced normalized normal constraint method,” *Structural and multidisciplinary optimization*, no. 36, pp. 537 – 546, 2008.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 124 – 141, 2002.
- [9] C. Fonseca and P. Fleming, “Genetic algorithms for multiobjective optimization: formulation, discussion an generalization.” *Proceedings of the fifth international conference on genetic algorithms.*, pp. 416 – 423, 1993.
- [10] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712 – 731, 2007.
- [11] C. A. C. Coello and G. B. Lamont, *Applications of Multi-Objective evolutionary algorithms*, advances in natural computation vol. 1 ed. World scientific publishing, 2004.
- [12] S. Srinivasan and S. Ramakrishnan, “Evolutionary multi objective optimization for rule mining: a review,” *Artificial Intelligence Review*, vol. 36, pp. 205–248, 2011, 10.1007/s10462-011-9212-3.
- [13] T. Aslam and A. Ng, “Multi-objective optimization for supply chain management: A literature review and new development,” in *Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on*, 2010, pp. 1 –8.

- [14] S. A. Mansouri, D. Gallear, and M. H. Askariyazad, “Decision support for build-to-order supply chain management through multiobjective optimization,” *International Journal of Production Economics*, vol. 135, no. 1, pp. 24 – 36, 2012.
- [15] M. Fadaee and M. Radzi, “Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review,” *Renewable and Sustainable Energy Reviews*, vol. 16, no. 5, pp. 3364 – 3369, 2012.
- [16] S. Fazlollahi, P. Mandel, G. Becker, and F. Maréchal, “Methods for multi-objective investment and operating optimization of complex energy systems,” *Energy*, vol. 45, no. 1, pp. 12 – 22, 2012.
- [17] Y. Sun, C. Zhang, L. Gao, and X. Wang, “Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects,” *The International Journal of Advanced Manufacturing Technology*, vol. 55, pp. 723–739, 2011.
- [18] C. Coello Coello, “Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science, J. Martínez-Trinidad, J. Carrasco-Ochoa, C. Ben-Youssef Brants, and E. Hancock, Eds. Springer Berlin / Heidelberg, 2011, vol. 6718, pp. 22–33.
- [19] A. Efstratiadis and D. Koutsoyiannis, “One decade of multi-objective calibration approaches in hydrological modelling: a review,” *Hydrological Sciences Journal*, vol. 55, no. 1, pp. 58–78, 2010.
- [20] P. Reed, D. Hadka, J. Herman, J. Kasprzyk, and J. Kollat, “Evolutionary multiobjective optimization in water resources: The past, present, and future,” *Advances in Water Resources*, vol. 51, no. 1, pp. 438 – 456, 2013.
- [21] N. Yusup, A. M. Zain, and S. Z. M. Hashim, “Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007-2011),” *Expert Systems with Applications*, vol. 39, no. 10, pp. 9909 – 9927, 2012.
- [22] K. Metaxiotis and K. Liagkouras, “Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review,” *Expert Systems with Applications*, vol. 39, no. 14, pp. 11 685 – 11 698, 2012.
- [23] G. Reynoso-Meza, X. Blasco, J. Sanchis, and M. Martínez, “Evolutionary algorithms for PID controller tuning: Current trends and perspectives (in spanish),” *Revista Iberoamericana de Automática e Informática Industrial*, vol. 10, no. 3, pp. 251 – 268, 2013.
- [24] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, “Controller tuning using evolutionary multi-objective optimisation: current trends and applications,” *Control Engineering Practice*, vol. -, no. -, pp. -, 20XX.
- [25] A. Lotov and K. Miettinen, “Visualizing the Pareto frontier,” in *Multiobjective Optimization*, ser. Lecture Notes in Computer Science, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Springer Berlin / Heidelberg, 2008, vol. 5252, pp. 213–243.

- [26] A. Inselberg, “The plane with parallel coordinates,” *The Visual Computer*, vol. 1, pp. 69–91, 1985.
- [27] X. Blasco, J. Herrero, J. Sanchis, and M. Martínez, “A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization,” *Information Sciences*, vol. 178, no. 20, pp. 3908 – 3924, 2008.
- [28] G. Reynoso-Meza, X. Blasco, J. Sanchis, and J. M. Herrero, “Comparison of design concepts in multi-criteria decision-making using level diagrams,” *Information Sciences*, vol. 221, pp. 124 – 141, 2013.
- [29] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, “Engineering applications of the self-organizing map,” *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.
- [30] P. Bonissone, R. Subbu, and J. Lizzi, “Multicriteria decision making (MCDM): a framework for research and applications,” *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 48 –61, 2009.
- [31] C. A. Mattson and A. Messac, “Pareto frontier based concept selection under uncertainty, with visualization,” *Optimization and Engineering*, vol. 6, pp. 85–115, 2005.
- [32] C. A. C. Coello, D. V. Veldhuizen, and G. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic press, 2002.
- [33] C. Coello, “Handling preferences in evolutionary multiobjective optimization: a survey,” in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, 2000, pp. 30 –37 vol.1.
- [34] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 2419 –2426.
- [35] A. Messac, “Physical programming: effective optimization for computational design.” *AIAA Journal*, vol. 34, no. 1, pp. 149–158, 1996.
- [36] D. Nagrath, M. Avila-Elchiver, F. Berthiaume, A. W. Tilles, A. Messac, and M. L. Yarmush, “Soft constraints-based multiobjective framework for flux balance analysis,” *Metabolic Engineering*, vol. 12, no. 5, pp. 429 – 445, 2010.
- [37] D. Cvetkovic and I. Parmee, “Preferences and their application in evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 42 –57, 2002.
- [38] M. Munro and B. Aouni, “Group decision makers’ preferences modelling within the goal programming model: An overview and a typology,” *Journal of Multi-Criteria Decision Analysis*, vol. 19, no. 3-4, pp. 169–184, 2012.
- [39] C. Fonseca and P. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms-I: A unified formulation,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 1, pp. 26 – 37, 1998.

- [40] R. Purshouse, C. Jalba, and P. Fleming, “Preference-driven co-evolutionary algorithms show promise for many-objective optimisation,” in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds. Springer Berlin / Heidelberg, 2011, vol. 6576, pp. 136 – 150, 10.1007/978-3-642-19893-9 10.
- [41] E. Fernandez, E. Lopez, F. Lopez, and C. A. C. Coello, “Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method,” *Information Sciences*, vol. 181, no. 1, pp. 44 – 56, 2011.
- [42] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, “A preference-based evolutionary algorithms for multi-objective optimization,” *Evolutionary Computation*, no. 3, pp. 411 – 436, 2009.
- [43] D. W. Corne and J. D. Knowles, “Techniques for highly multiobjective optimisation: some nondominated points are better than others,” in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ser. GECCO ’07. New York, NY, USA: ACM, 2007, pp. 773 – 780.
- [44] Y. Wang and Y. Yang, “Particle swarm optimization with preference order ranking for multi-objective optimization,” *Information Sciences*, vol. 179, no. 12, pp. 1944 – 1959, 2009.
- [45] B. Qu and P. Suganthan, “Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection,” *Information Sciences*, vol. 180, no. 17, pp. 3170 – 3181, 2010, including Special Section on Virtual Agent and Organization Modeling: Theory and Applications.
- [46] J. Bader and E. Zitzler, “Hype: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, no. 1, pp. 45 – 76, 2011.
- [47] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, “Reference point based multi-objective optimization using evolutionary algorithms,” *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.
- [48] J. Sanchis, M. A. Martínez, X. Blasco, and G. Reynoso-Meza, “Modelling preferences in multiobjective engineering design,” *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 1255 – 1264, 2010.
- [49] M. Martínez, J. Sanchis, and X. Blasco, “Multiobjective controller design handling human preferences,” *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 927 – 938, 2006.
- [50] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, “Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning,” in *Applications of Evolutionary Computation*. Springer, 2010, pp. 532–541.
- [51] G. Reynoso-Meza, “Design, coding and implementation of a multiobjective optimization algorithm based on differential evolution with spherical pruning: applications for system identification and controller tuning.” Master’s thesis, Universitat Politècnica de València., 2009. [Online]. Available: <http://personales.upv.es/gilreyme/Tesina/files/TesinaGRM.zip>

- [52] R. Storn and K. Price, “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341 – 359, 1997.
- [53] E. Mezura-Montes, M. Reyes-Sierra, and C. Coello, “Multi-objective optimization using differential evolution: A survey of the state-of-the-art,” *Advances in Differential Evolution*, no. SCI 143, pp. 173 – 196, 2008.
- [54] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1 –28, 2010.
- [55] J. Branke, H. Schmeck, K. Deb, and M. Reddy S, “Parallelizing multi-objective evolutionary algorithms: cone separation,” in *Evolutionary Computation, 2004. CEC2004. Congress on, 2004*.
- [56] O. Kramer and P. Koch, “Rake selection: A novel evolutionary multi-objective optimization algorithm,” in *KI 2009: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, B. Mertsching, M. Hund, and Z. Aziz, Eds. Springer Berlin / Heidelberg, 2009, vol. 5803, pp. 177–184.
- [57] L. Batista, F. Campelo, F. Guimarães, and J. Ramírez, “Pareto cone  $\epsilon$ -dominance: Improving convergence and diversity in multiobjective evolutionary algorithms,” in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds. Springer Berlin / Heidelberg, 2011, vol. 6576, pp. 76–90.
- [58] R. Storn, “Differential evolution research: Trends and open questions,” U. K. C. (Ed.), Ed. Springer, Heidelberg, 2008, vol. LNCS 143, pp. 1 – 31.
- [59] X. Blasco, S. García-Nieto, and G. Reynoso-Meza, “Autonomous trajectory control of a quadcopter vehicle. simulation and evaluation,” *Revista Iberoamericana de Automática e Informática Industrial*, vol. 9, no. 2, pp. 194 – 199, 2012.
- [60] X. Blasco, G. Reynoso-Meza, and S. García-Nieto, “Resultados del concurso de ingeniería de control 2012 y convocatoria 2013,” *Revista Iberoamericana de Automática e Informática Industrial*, vol. 10, no. 2, pp. 240 – 244, 2013.
- [61] K.-B. Lee and J.-H. Kim, “Multi-objective particle swarm optimization with preference-based sorting,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 2506–2513.
- [62] I. Kaliszewski, J. Miroforidis, and D. Podkopaev, “Interactive multiple criteria decision making based on preference driven evolutionary multiobjective optimization with controllable accuracy,” *European Journal of Operational Research*, vol. 216, no. 1, pp. 188–199, 2012.
- [63] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. Herrero, “Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 1551 –1556.

- [64] K. Deb, L. Tiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” Institut für Technische Informatik und Kommunikationsnetze, ETH Zurich, Tech. Rep. TIK No. 112, Feb. 2001.
- [65] B. Wie and D. S. Bernstein, “Benchmark problems for robust control design,” *Journal of Guidance, Control and Dynamics.*, no. 15, pp. 1057 – 1059, 1992.
- [66] E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervós, J. Bullinaria, J. Rowe, P. Tino, A. Kabán, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3242, pp. 832–842.
- [67] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117 – 132, 2003.
- [68] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3 – 18, 2011.