We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Evolutionary Multi-Objective Robust Optimization

J. Ferreira*, C. M. Fonseca**, J. A. Covas* and
A. Gaspar-Cunha*
I3N, University of Minho, Guimarães,
** Centre for Intelligent Systems, University of Algarve, Faro,
Portugal

1. Introduction

Most practical engineering optimization problems are multi-objective, *i.e.*, their solution must consider simultaneously various performance criteria, which are often conflicting. Multi-Objective Evolutionary Algorithms (MOEAs) are particularly adequate for solving these problems, as they work with a population (of vectors or solutions) rather than with a single point (Schaffer, 1984; Fonseca & Fleming, 1993; Srinivas & Deb, 1995; Horn et al., 1994; Deb et al., 2002; Zitzler et al., 2001; Knowles & Corne, 2000; Gaspar-Cunha et al. 2004). This feature enables the creation of Pareto frontiers representing the trade-off between the criteria, simultaneously providing a link with the decision variables (Deb, 2001, Coello et al., 2002). Moreover, since in real applications small changes of the design variables or of environmental parameters may frequently occur, the performance of the optimal solution (or solutions) should be only slightly affected by these, *i.e.*, the solutions should also be robust (Ray, 2002; Jin & Branke, 2005). The optimization problems involving unmanageable stochastic factors can be typified as (Jin & Branke, 2005): i) those where the performance is affected by noise originated by sources such as sensor measurements and/or environmental parameters (Wiesmann et al., 1998; Das, 1997); ii) those where the design variables change after the optimal solution has been found (Ray, 2002; Tsutsui & Ghosh, 1997; Chen et al., 1999); iii) problems where the process performance is estimated by an approximation to the real value; iv) and those where the performance changes with time, which implies that the optimization algorithm must be updated continuously. This text focuses exclusively problems of the second category.

Given the above, optimization algorithms should determine the solutions that simultaneously maximize performance and guarantee satisfactory robustness, but the latter is rarely included in traditional algorithms. As robustness and performance can be conflicting, it is important to know their interdependency for each optimization problem. A robustness analysis should be performed as the search proceeds and not after, by introducing a robustness measure during the optimization. Robustness can be studied either by replacing the original objective function by an expression measuring both the performance and the expectation of each criterion in the vicinity of a specific solution, or by inserting an additional optimization criterion assessing robustness in addition to the original

criteria. As will be demonstrated in the next sections, in the first situation the role of the optimization algorithm is to find the solution that optimizes the expectation (in the vicinity of the solutions considered) of the original criterion (or criteria), while in the second case a trade-off between the original criteria and the robustness measure is obtained (Jin & Sendhoff, 2003).

In single objective (or criterion) optimization, the best solution is the one that satisfies simultaneously performance and robustness. Robust single objective optimization has been applied to various engineering fields and using different optimization methodologies (Ribeiro & Elsayed, 1995; Tsutsui & Ghosh, 1997; Das, 1997; Wiesmann et al., 1998; Du & Chen, 1998; Chen et al. 1999; Ray, 2002; Arnold & Beyer, 2003; Sorensen, 2004). However, only recently robustness analysis has been extended to Multi-Objective Optimization Problems (MOOP) (Kouvelis & Sayin 2002; Bagchi, 2003; Jin & Sendhoff, 2003; Kazancioglu et al., 2003; Gaspar-Cunha & Covas, 2005; Ölvander, 2005; Guanawan & Azarm, 2005; Deb & Gupta, 2006; Paenke et al., 2006; Barrico & Antunes, 2006; Moshaiov & Avigrad, 2006; Gaspar-Cunha & Covas, 2008). Depending on the type of Pareto frontier, the aim can be: i) to locate the optimal Pareto front's most robust section (Deb & Gupta, 2006; Gaspar-Cunha & Covas, 2008) and/or ii) in the case of a multimodal problem, to find the most robust Pareto frontier, and not only the most robust region of the optimal Pareto frontier (Guanawan & Azarm, 2005; Deb & Gupta, 2006).

An important question arising from MOOP is the choice of the (single) solution to be used on the real problem under study (Ferreira et al., 2008). Generally, to select a solution from the pool of the available ones, the Decision Maker (DM) characterizes the relative importance of the criteria and subsequently applies a decision methodology. The use of a weighted stress function approach (Ferreira et al., 2008) is advantageous, as it enables the DM to define the extension of the optimal Pareto frontier to be obtained, via the use of a dispersion parameter. This concept could be adapted by taking into account robustness and not the relative criteria importance.

Consequently, this work aims to discuss robustness assessment during multi-objective optimization using a MOEA, namely in terms of the identification of the robust region (or regions) of the optimal Pareto frontier. The text is organized as follows. In section 2, robustness concepts will be presented and extended to multi-objective optimization. The multi-objective evolutionary algorithm used and the corresponding modifications required to take robustness into account will be described and discussed in section 3. The performance of the robustness measures will be evaluated in section 4 via their application to several benchmark multi-objective optimization problems. Finally, the main conclusions are summarized in section 5.

2. Robustness concepts

2.1 Single objective optimization
A single objective optimization can be formulated as follows:

$$
\begin{aligned}
& \max_{x_l} && f(x_l) && l = 1, \cdots, L \\
& \text{subject to} && g_j(x_l) = 0 && j = 1, \cdots, J \\
& && h_k(x_l) \geq 0 && k = 1, \cdots, K \\
& && x_{l,min} \leq x_l \leq xl,_{max}
\end{aligned}
\tag{1}
$$

where x_l are the L parameters (or design vectors) x_1, x_2, ..., x_L, g_j and h_k are the J equality ($J\geq 0$) and K inequality ($K\geq 0$) constraints, respectively, and $x_{l,min}$ and $x_{l,max}$ are the lower and upper limits of the parameters.

The most robust solution is that for which the objective function f is less sensitive to variations of the design parameters x_l. Figure 1 shows the evolution of the objective function $f(x_1,x_2)$ (to be maximized) against the design parameter x_1, when another factor and/or the design parameter x_2 changes slightly from x_2' to x_2''. Solution S_2 is less sensitive than solution S_1 to variations of x_2, since the changes in the objective function are less significant (Δf_2 and Δf_1 for S_2 and S_1, respectively) and, consequently, it can be considered as the most robust solution (taking into consideration that here robustness is measured only as a function of changes occurring in the objective function). On the other hand, since S_1 is more performing than S_2, a balance between performance (or fitness) of a solution and its robustness has to be done. In spite of its lower fitness, solution S_2 is the most robust and would be the selected one by an optimization algorithm (Guanawan & Azarm, 2005; Gaspar-Cunha & Covas, 2005; Deb & Gupta, 2006; Paenke et al., 2006; Gaspar-Cunha & Covas, 2008).
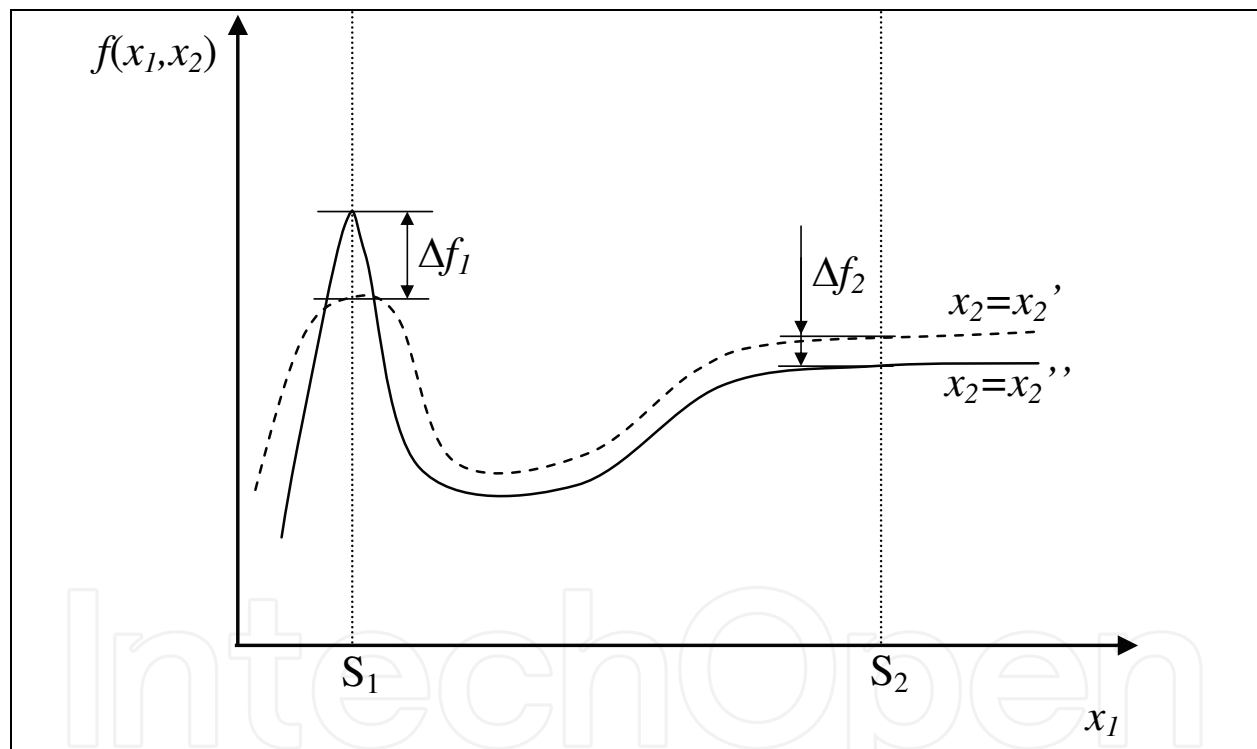

Fig. 1. Concept of robustness in the case of a single objective function

Two major approaches have been developed in order to deal with robustness in an optimization process (Ray, 2002; Jin & Sendhoff, 2003; Gaspar-Cunha & Covas, 2005; Deb & Gupta, 2006; Gaspar-Cunha & Covas, 2008):

- **Expectation measure**: the original objective function is replaced by a measure of both its performance and expectation in the vicinity of the solution considered. Figure 2 illustrates this method. Figure 2-A shows that in function $f(x)$, having five different peaks, the third is the most robust, since fitness fluctuations around its maximum are smaller. However, most probably, an optimization algorithm would select the first peak. An expectation measure

takes this fact into account by replacing the original function by another such as that illustrated in Figure 2-B. Now, if a conventional optimization is performed using this new function, the peak selected (peak three) will be the most robust. Various types of expectation measures have been proposed in the literature (Tsutsui & Ghosh, 1997; Das, 1997; Wiesmann et al., 1998; Jin & Sendhoff, 2003; Gaspar-Cunha & Covas, 2005; Deb & Gupta, 2006; Gaspar-Cunha & Covas, 2008).
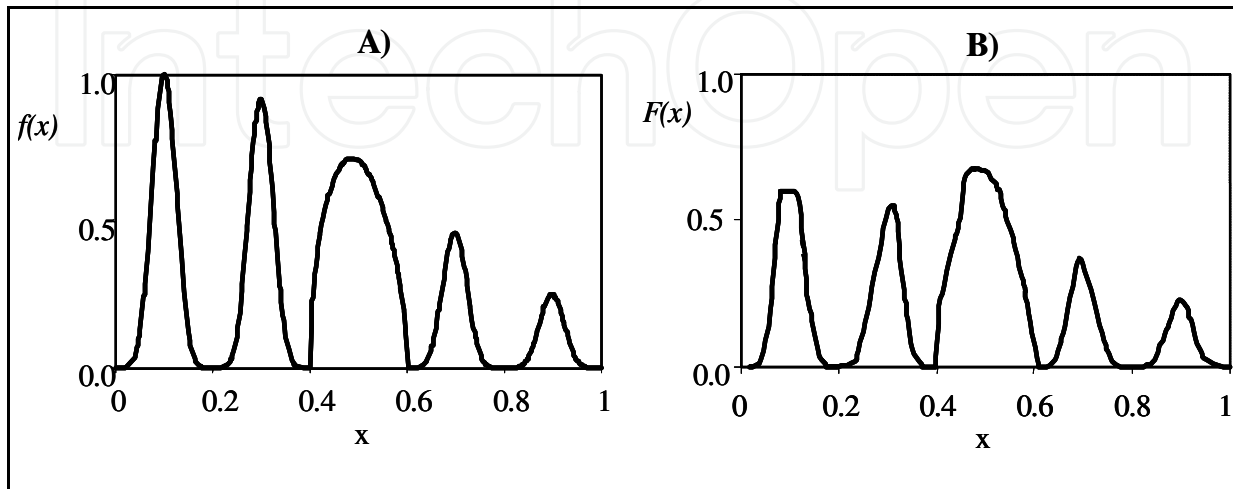

Fig. 2. Expectation measure for a single objective function

- **Variance measure**: An additional criterion is appended to the objective function to measure the deviation of the latter around the vicinity of the design point. Variance measures take only into account function deviations, ignoring the associated performance. Thus, in the case of a single objective function, the optimization algorithm must perform a two-criterion optimization, one concerning performance and the other robustness (Jin & Sendhoff, 2003; Gaspar-Cunha & Covas, 2005; Deb & Gupta, 2006; Gaspar-Cunha & Covas, 2008).

Deb & Gupta (2006) denoted the above two approaches as type I and II, respectively. The performance of selected expectation and variance measures was evaluated in terms of their capacity to detect robust peaks (Gaspar-Cunha & Covas, 2008), by assessing such features as: i) easy application to problems where the shape of the objective function is not known a priori, ii) capacity to define robustness regardless of that shape, iii) independence of the algorithm parameters, iv) clear definition of the function maxima in the Fitness versus Robustness Pareto representation, and v) efficiency. The best performance was attained when the following variance measure was used:

$$f_i^R = \frac{1}{N'}\sum_{j=0}^{N}\left|\frac{\tilde{f}(x_j)-\tilde{f}(x_i)}{x_j-x_i}\right|, \qquad d_{i,j} < d_{max} \tag{2}$$

where the robustness of individual i is defined as the average value of the ratio of the difference between the normalized fitness of individual i, $\tilde{f}(x_i)$, and that of its neighbours (j), over the distance separating them. In this expression, $\tilde{f}(x_i) = \frac{f(x_i)-f_{min}}{f_{max}-f_{min}}$ for maximization

and $\tilde{f}(x_i) = 1 - \dfrac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}$ for minimization of the objective function $f(x_i)$, with f_{max} and f_{min} representing the limits of its range of variation, N' is the number of population individuals whose Euclidian distance between points i and j ($d_{i,j}$) is lower than d_{max} (*i.e.*, $d_{i,j} < d_{max}$):

$$d_{i,j} = \sqrt{\sum_{m=1}^{M} \left(x_{m,j} - x_{m,i}\right)^2} \tag{3}$$

and M is the number of criteria. The smaller $f^R{}_i$, the more robust the solution is.

2.2 Extending robustness to multiple objectives
In a multi-objective optimization various objectives, often conflicting, co-exist:

$$
\begin{aligned}
&\max_{x_l} && f_m(x_l) && l = 1, \cdots, L \qquad m = 1, \cdots, M \\
&\text{subject to} && g_j(x_l) = 0 && j = 1, \cdots, J \\
& && h_k(x_l) \geq 0 && k = 1, \cdots, K \\
& && x_{l,min} \leq x_l \leq xl,_{max}
\end{aligned}
\tag{4}
$$

where f_m are the M objective functions of the L parameters (or design vectors) $x_1, x_2, \ldots, x_L$ and g_j and h_k are the J equality ($J \geq 0$) and K inequality ($K \geq 0$) constraints, respectively.

The application of a robustness analysis to MOOPs must consider all the criteria simultaneously. As for single objective, a multi-objective robust solution must be less sensitive to variations of the design parameters, as illustrated in Figure 3. The figure shows that the same local perturbation on the parameters space (x_1, x_2) causes different behaviours of solutions I and II. Solution I is more robust, as the same perturbations on the parameters space causes lower changes on the objective space. Each of the Pareto optimal solutions must be analysed in what concerns robustness, *i.e.*, its sensitivity to changes on the design parameters. Since robustness must be assessed for every criterion, the combined effect of changes in all the objectives must be considered simultaneously and used as a measure of robustness.
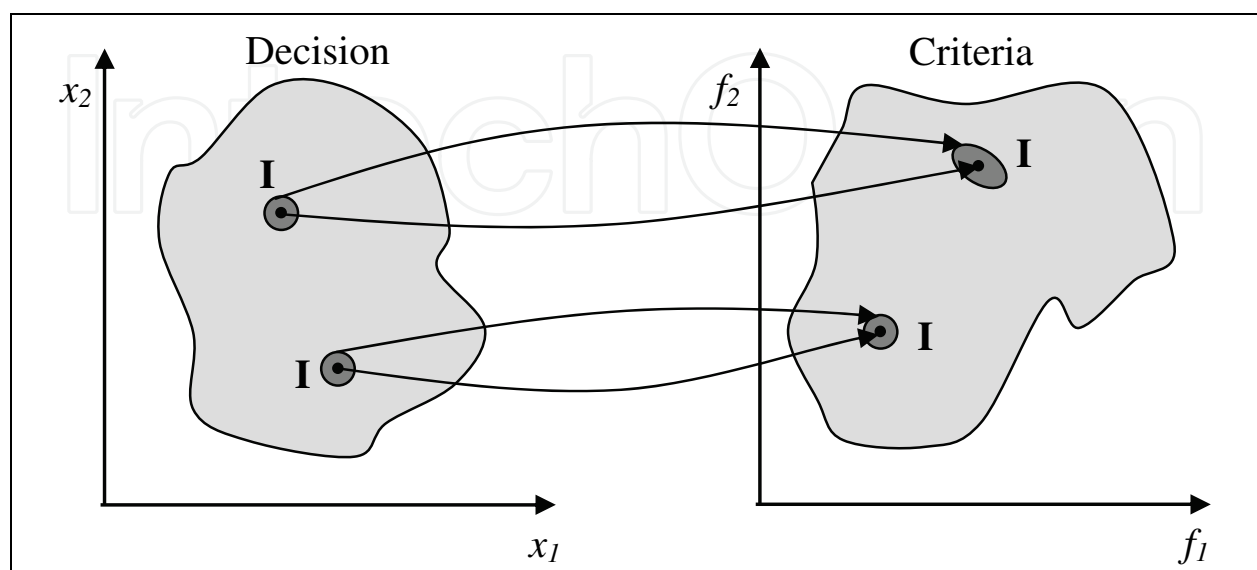

Fig. 3. Concept of robustness for multi-objective functions

In multi-objective robust optimization the aim is to obtain a set of Pareto solutions that are, at the same time, multi-objectively robust and Pareto optimal. As shown in Figure 4, different situations may arise (Guanawan & Azarm, 2005; Deb & Gupta, 2006):

1. All the solutions on the Pareto-optimal frontier are robust (Figure 4-A);
2. Only some of the solutions belonging to the Pareto-optimal frontier are robust (Figure 4-B);
3. The solutions belonging to the Pareto-optimal frontier are not robust, but a robust Pareto frontier exists (Figure 4-C);
4. Some of the robust solutions belong to the Pareto-optimal frontier, but others do not (Figure 4-D).
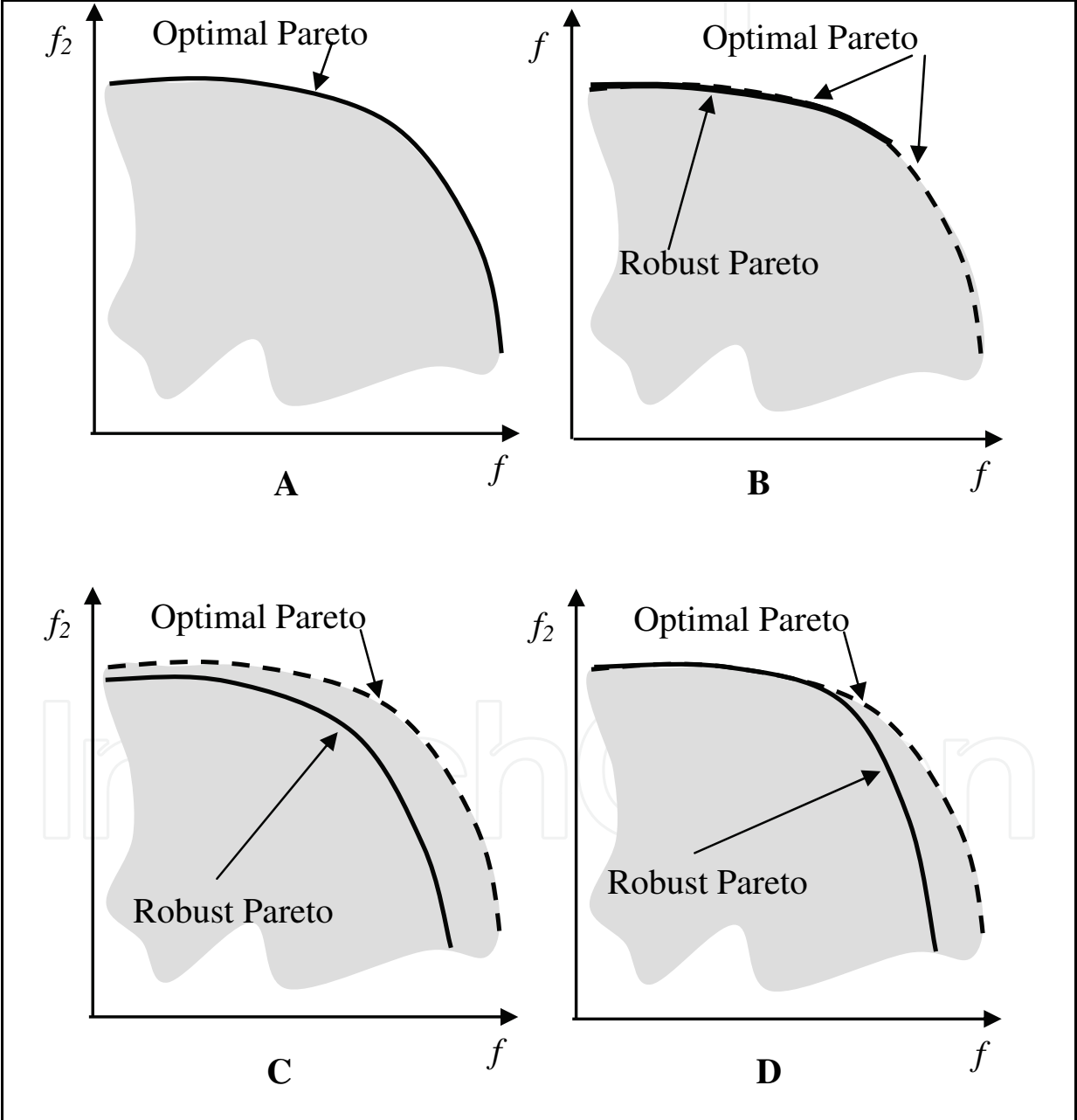

Fig. 4. Optimal Pareto frontier versus robust Pareto frontier

All the above situations should be taken into consideration by a resourceful optimization algorithm. When the DM is only interested in the most robust section of the optimal Pareto frontier (see Figure 5), this can be done by using, for example, the dispersion parameter referred above.
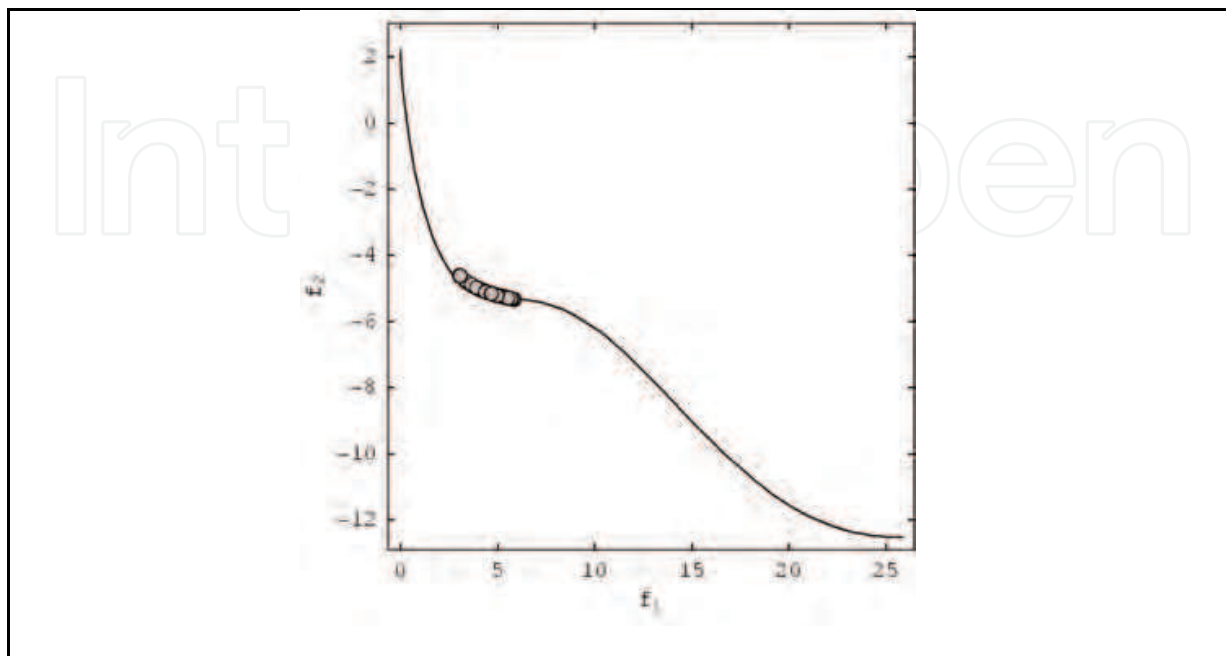

Fig. 5. Robust region of the optimal Pareto frontier (Test Problem 1, see below)

3. Multi-objective optimization

3.1 Multi-Objective Evolutionary Algorithms (MOEAs)

Multi-Objective Evolutionary Algorithms (MOEAs) are an efficient tool to deal with the above type of problems, since they are able to determine in a single run the optimal Pareto front. For that reason, they have been intensively used in the last decade (Fonseca & Fleming, 1998; Deb, 2001, Coello et al., 2002; Gaspar-Cunha & Covas, 2004).

A MOEA must provide the homogeneous distribution of the population along the Pareto frontier, together with improving the solutions along successive generations. Usually, a fitness assignment operator is applied to guide the population towards the Pareto frontier using a robust and efficient multi-objective selection method, as well as a density estimation operator to maintain the solutions dispersed along the Pareto frontier, as it is able to take into account the proximity of the solutions. Moreover, in order to prevent fitness deterioration along the successive generations, an archiving process is introduced by maintaining an external population where the best solutions found sequentially are kept and periodically incorporated into the main population.

The Reduced Pareto Set Genetic Algorithm with elitism (RPSGAe) will be adopted in this chapter (Gaspar-Cunha et al., 1997), although some changes in its working mode have to be implemented in order to take into account the robustness procedure proposed. RPSGAe is able to distribute the solutions uniformly along the Pareto frontier, its performance having been assessed using benchmark problems and statistical comparison techniques. The method starts by sorting the population individuals in a number of pre-defined ranks using a clustering technique, thus reducing the number of solutions on the efficient frontier while

maintaining intact its characteristics (Gaspar-Cunha & Covas, 2004). Then, the individuals' fitness is calculated through a ranking function. With the aim of incorporating this technique, the traditional GA was modified as follows (Gaspar-Cunha & Covas, 2004):

1. Random initial population (internal)
2. Empty external population
3. while not Stop-Condition do

      a- Evaluate internal population
      b- **Calculate expectation and/or robustness measures**
      c- **Calculate niche count (mi)**
      d- Calculate the Ranking of the individuals using the RPSGAe
      e- **Calculate the global Fitness ($\tilde{F}(i)$)**

      f- Copy the best individuals to the external population
      g- if the external population becomes full

            Apply the RPSGAe to this population
            Copy the best individuals to the internal population

      end if
      h- Select the individuals for reproduction
      i- Crossover
      j- Mutation
end while

As described above, the calculations start with the random definition of an internal population of size *N* and of an empty external population of size *Ne*. At each generation, a fixed number of the best individuals (that was obtained by reducing the internal population with the clustering algorithm), is copied to an external population (Gaspar-Cunha et al., 1997). The process is repeated until the external population becomes complete. Then, the RPSGAe is applied to sort the individuals of this population, and a pre-defined number of the best individuals is incorporated in the internal population, by replacing the lowest fitness individuals. Detailed information on this algorithm can be found elsewhere (Gaspar-Cunha & Covas, 2004; Gaspar-Cunha, 2000).

3.2 Introducing robustness in MOEAs

Three additional steps must be added on to the RPSGAe presented above, to comprise robustness estimation. They consist of a computation of robustness measures (taking into account the dispersion parameter), a niche count and the determination of the global fitness, yielding the general flowchart of Figure 7. The dispersion parameter (ε') quantifies the extension of the robust section to be obtained (see Figure 5). This parameter can be defined by the DM and ranges between 0, when a single solution is to be obtained, and 1, when the entire optimal Pareto frontier is to be obtained. In order to consider the influence of the dispersion parameter (ε'), the way how the indifference limits ($\tilde{L}_j$) and the distances between the solutions ($\tilde{D}_{j,k}$) are defined in the RPSGAe algorithm was also changed (see Gaspar-Cunha & Covas, 2004), the following equations being used:

$$\tilde{L}_j = L_j \times \left(1 + \frac{2}{(\max R - \min R)}\right)^{\frac{1-\varepsilon'}{\varepsilon'}} \tag{5}$$

$$\tilde{D}_{j,k} = D_{j,k} \times \left(1 + \frac{1}{R(\text{ind}_{k+1})}\right)^{\frac{1-\varepsilon'}{\varepsilon'}} \tag{6}$$

Here, max R and min R are the maximum and the minimum values of the robustness found for each generation, respectively, L_i are the indifference limits for criterion i, $D_{j,k}$ is the difference between the criterion value of solutions j and k, $R(ind_{k+1})$ is the robustness measure of the individual located in position $k+1$ after the population was ordered by criterion j. The robustness measure is calculated by Equation 2, thus when R increases the robustness of the solution decreases. In these equations, the dispersion parameter (ε') plays an important role. If $\varepsilon'=1$, equations 5 and 6 are reduced to L_i and $D_{i,j}$, respectively, and the algorithm will converge for the entire robust Pareto frontier. Otherwise, when ε' decreases, the size of the robust Pareto frontier decreases as well. In a limiting situation, *i.e.*, when ε' is approximately nil, a single point is obtained. Figure 8 shows curves of $\tilde{L}_j / L_j$ and $\tilde{D}_{j,k} / D_{j,k}$ ratios against the dispersion parameter, for different values of R (2.0, 0.5 and 0.1).
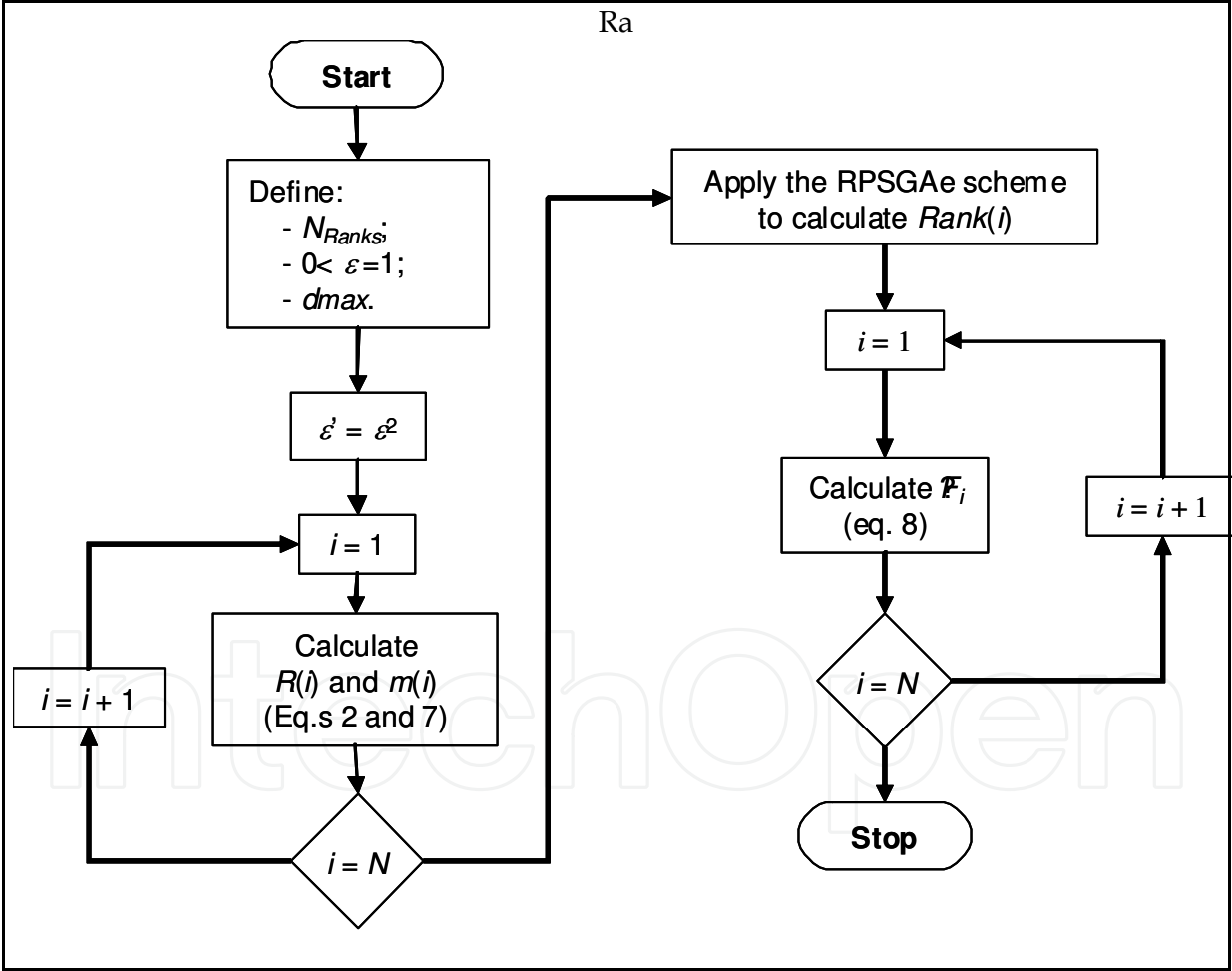

Fig. 7. Flowchart of the robustness routine

Ratio $\tilde{D}_{j,k} / D_{j,k}$ is given by $\left(1 + \frac{1}{R(\text{ind}_{k+1})}\right)^{\frac{1-\varepsilon'}{\varepsilon'}}$ (see equation 6). Thus, at constant R, when ε' decreases means that influence of the difference between the value of solutions j and k (*i.e.*,

$D_{j,k}$) on $\tilde{D}_{j,k}$ diminishes. Therefore, for small values of the dispersion parameter, the attribution of the fitness by the RPSGAe algorithm is made almost exclusively by the value of the robustness of the solutions and not by taking into account the distance between them. This procedure avoids that robust solutions are eliminated during the consecutive generations in case they are next to each other. An identical analysis can be made for different robustness values (R in Figure 8). When R increases (*i.e.*, when the robustness decreases) the value of $\tilde{D}_{j,k} / D_{j,k}$ must decreases in order to produce the same result. The same reasoning applies to the $\tilde{L}_j / L_j$ ratio.
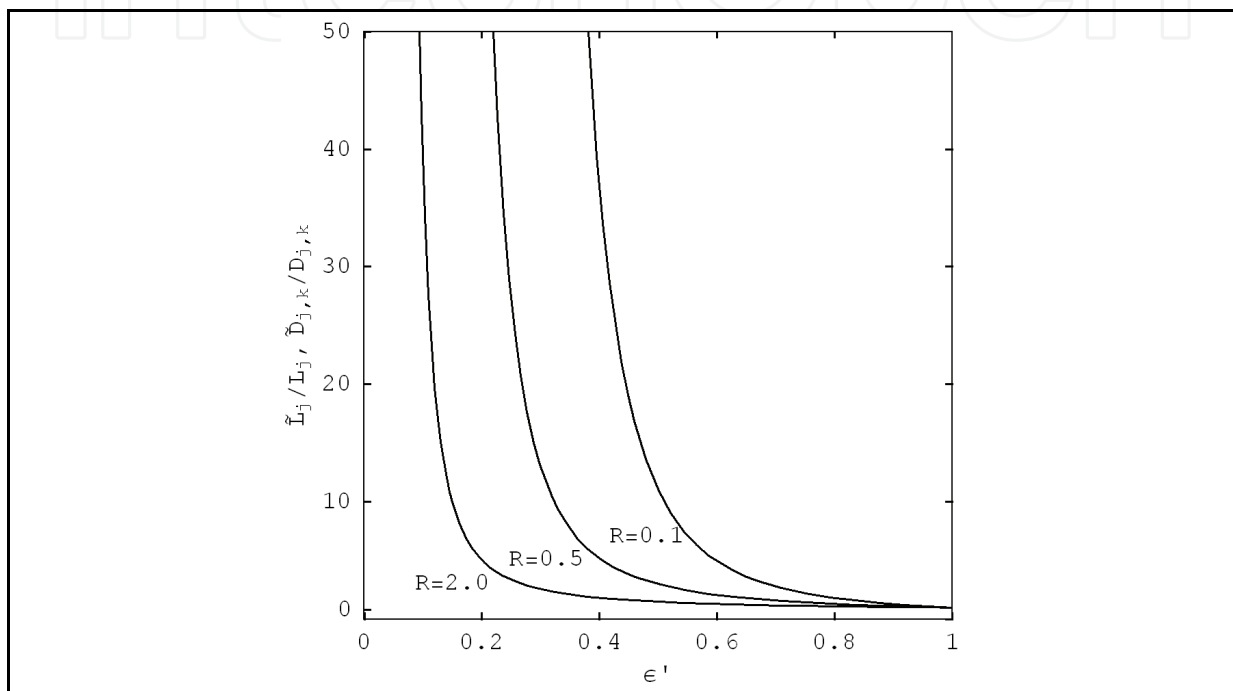

Fig. 8. Shape of the curves of $\tilde{L}_j / L_j$ and $\tilde{D}_{j,k} / D_{j,k}$ rates as a function of the dispersion parameter for different R values

The niche count was considered using a sharing function (Goldberg & Richardson, 1987):

$$m(i) = \sum_{j=1}^{N} sh(d_{ij}) \tag{7}$$

where $sh(d_{ij})$ is related to individual i and takes into account its distance to all its neighbours j (d_{ij}).
Finally, the global fitness was calculated using the following equation:

$$\tilde{F}(i) = Rank(i) + (1 - \varepsilon')\frac{R(i)}{R(i)+1} + \varepsilon'\frac{m(i)}{m(i)+1} \tag{8}$$

In conclusion, the following calculation steps must be carried out (see Figure 7):
1. The robustness routine starts with the definition of the number of ranks (N_{ranks}), the span of the Pareto frontier to be obtained ($\varepsilon \in [0,1]$) and the maximum radial distance to each solution to be considered in the robustness calculation (d_{max});

2. To reduce the sensitivity of the algorithm to small values of the objective functions, the dispersion parameter is changed as $\varepsilon' = \varepsilon^2$;
3. For each individual, i, robustness , $R(i)$, and niche count, $m(i)$, are determined using equations 2 and 7, respectively;
4. The RPSGAe algorithm is applied, with the modifications introduced by equations 5 and 6, to calculate $Rank(i)$;
5. For each solution, i, the new fitness is calculated using equation 8.

4. Results and discussion

4.1 Test problems

The robustness methodology presented in the previous sections will be tested using the 7 Test Problems (TP) listed below, each of different type and with distinctive Pareto frontier characteristics. Each TP is presented in terms of its creator, aim, number of decision parameters, criteria and range of variation of the decision parameter.

TP 1 and 2 are simple one parameter problems, the first having one region with higher robustness, while the second contains three such regions. TP 3 to TP5 are complex MOOPs with 30 parameters each, and two criteria. TP3 and TP4 have a single region with higher robustness and the Pareto frontier is convex and concave, respectively. TP5 has a discontinuous Pareto frontier with a single region with higher robustness. TP 6 and TP7 are the three criteria version of TP1 and TP4, respectively.

Three studies will be performed, to determine: i) the effect of the RPSGAe algorithm, *i.e.*, N_{ranks}, and d_{max}; ii) the effect of the value of the dispersion parameter and iii) the performance of the robustness methodology for different type of problems.

The RPSGAe algorithm parameters utilized are the following: N_{ranks} = 20 (the values of 10 and 30 were also used for the first study), d_{max} = 0.008 (0.005 and 0.03 were also tried in the first study), indifference limits equal to 0.1 for all criteria, SBX real crossover operator with an index of 10 and real polynomial mutation operator with and index of 20.

TP 1: $x \in [-2;6]$; Minimize; $L=1$; $M=2$.

$$f_1(x) = x^2$$
$$f_2(x) = e^{|x|-5} + (6/5)\cos(2x) - 2,7x + 1$$

(9)

TP 2: $x \in [0;5]$; Maximize; $L=1$; $M=2$.

$$f_1(x) = x$$
$$f_2(x) = -5x + \cos(4x)$$

(10)

TP 3 (ZDT1): $x_i \in [0;1]$; Minimize; $L=30$; $M=2$; Deb, Pratapat et al., 2002.

$$f_1(x_1) = x_1$$
$$f_2(x_2, \cdots, x_L) = g(x) \times \left(1 - \sqrt{f_1(x_1)/g(x)}\right)$$

(11)

$$\text{with, } g(x) = 1 + 9 \frac{\sum_{l=2}^{L} x_l}{L-1}$$

TP 4 (ZDT2): $x_i \in [0;1]$; Minimize; L=30; M=2; Deb, Pratapat et al., 2002.

$$f_1(x_1) = x_1$$
$$f_2(x_2, \cdots, x_L) = g(x) \times \left(1 - \left(\frac{f_1(x_1)}{g(x)} \right)^2 \right) \tag{12}$$
$$\text{with, } g(x) = 1 + 9 \frac{\sum_{l=2}^{L} x_l}{L-1}$$

TP 5 (ZDT3): $x_i \in [0;1]$; Minimize; L=30; M=2; Deb, Pratapat et al., 2002.

$$f_1(x_1) = x_1$$
$$f_2(x_2, \cdots, x_L) = g(x) \times \left(1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right) \tag{13}$$
$$\text{with, } g(x) = 1 + 9 \frac{\sum_{l=2}^{L} x_l}{L-1}$$

TP 6: $x_1 \in [0;2\pi]$; $x_2 \in [0;5]$; Minimize; L=2; M=3.

$$f_1(x) = sin(x_1).g(x_2)$$
$$f_2(x) = cos(x_1).g(x_2)$$
$$f_3(x) = x_2^2 \tag{14}$$
$$\text{with, } g(x_2) = e^{x_2(x_2-5)} - \frac{6}{5} sin(2x_2) - 2,7x_2 - 1$$

TP 7 (DTLZ2): $x_i \in [0;1]$; Minimize; L=12; M=3; Deb, Thiele et al., 2002.

$$f_1(x) = (1 + g(x)).cos\left(x_1 \frac{\pi}{2} \right).cos\left(x_2 \frac{\pi}{2} \right)$$
$$f_2(x) = (1 + g(x)).cos\left(x_1 \frac{\pi}{2} \right).sin\left(x_2 \frac{\pi}{2} \right) \tag{15}$$
$$f_3(x) = (1 + g(x)).sin\left(x_1 \frac{\pi}{2} \right)$$
$$\text{with, } g(x) = \sum_{i=3}^{L} (x_i - 0.5)^2$$

4.2 Effect of the RPSGAe parameters

Figure 9 compares the results obtained with the robustness procedure for TP 1 and TP4, using different values of the parameter. The line indicates the optimal Pareto frontier and the dots identify the solutions obtained with the new procedure. As shown, the algorithm is able to produce good results independently of the value of N_{ranks} (hence, in the remaining of this study N_{ranks} was set as 20).

Similar conclusions were obtained for d_{max} parameter - Figure 10, so d_{max} was kept equal to 0.008.
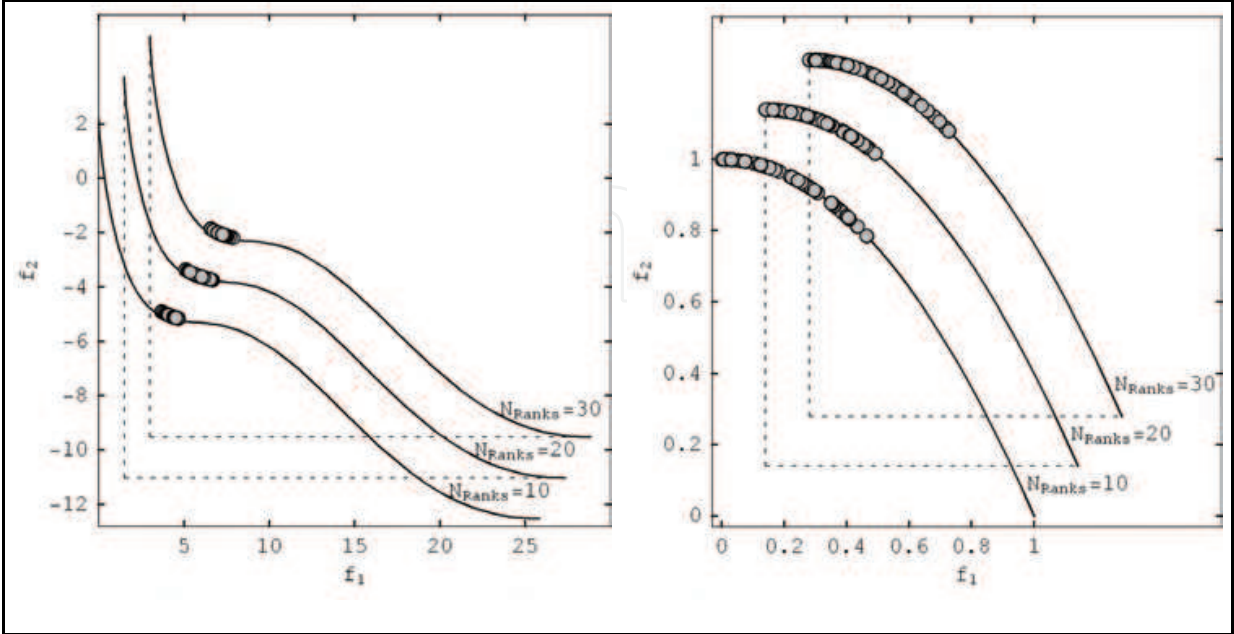

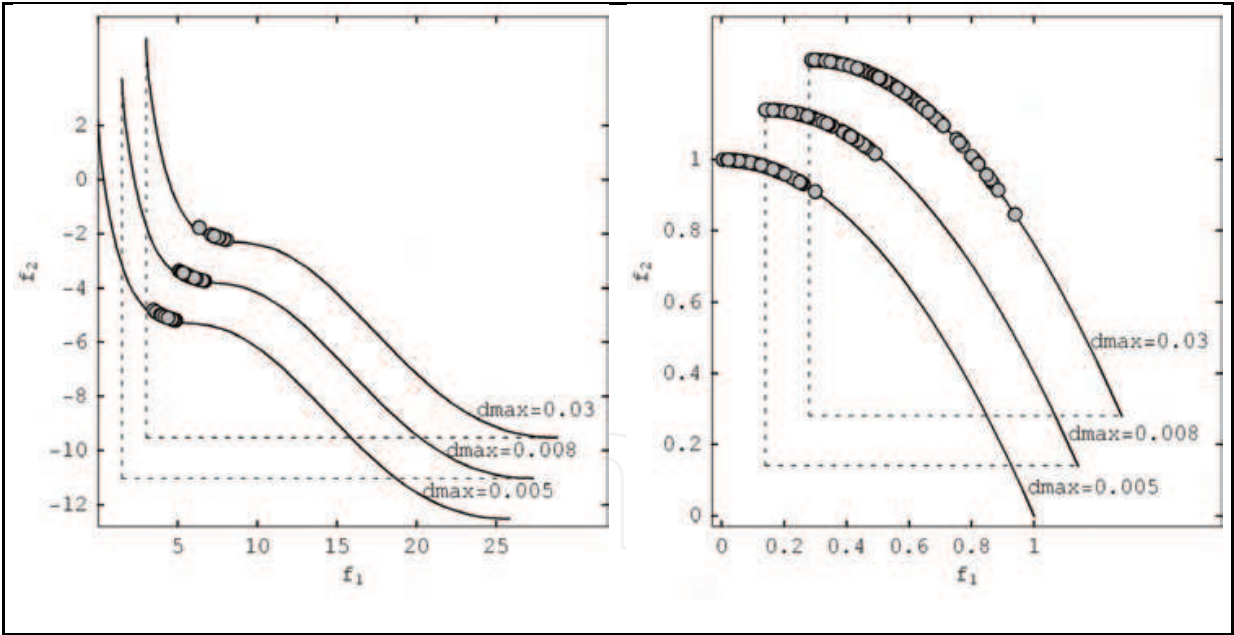Fig. 9. Influence of N_{ranks} parameter for TP1 and TP4

Fig. 10. Influence of d_{max} parameter for TP1 and TP4

4.3 Effect of the dispersion parameter

The aim of the dispersion parameter is to provide the Decision Maker with the possibility of choosing different sizes of the optimal/robustness Pareto frontier. Figure 11 shows the results obtained for TP1 using different values of that parameter, identical outcomes having been observed for the remaining test problems. The methodology seems to be sensitive to the variation on the dispersion parameter, which is a very positive feature.

4.4 Effect of the type of problem

The results obtained for TP2 to TP7, using $\varepsilon = 0.1$, are presented in Figure 12. The algorithm is able to deal with the various types of test problems proposed. TP2 is a difficult test problem due to the need to converge to the three different sections with the same robustness. TP3 and TP4 show that the algorithm proposed can converge to the most robust region even for problems with 30 parameters or of discontinuous nature. Finally, TP6 and TP7 show that the methodology proposed is able to deal with more than two dimensions with a good convergence, which is not generally the case for current optimization algorithms available.

5. Conclusions

This work presented and tested an optimization procedure that takes into account robustness in multi-objective optimization. It was shown that the method is able to deal with different types of problems and with different degrees of complexity.

The extension of the robust Pareto frontier can be controlled by the Decision Maker by making use of a dispersion parameter. The effectiveness of this parameter was demonstrated in a number of test problems.

6. References

Arrold, D.V. & Beyer, H.-G. (2003). A Comparison of Evolution Strategies with Other Direct Search Methods in the Presence of Noise, *Computational Optimization and Applications*, Vol. 24, No. 1 (2003) 135-159

Bagchi, T.P. (2003). Multiobjective Robust Design by Genetic Algorithms, *Materials and Manufacturing Processes*, Vol. 18, No. 3 (2003) 341-354

Barrico, C. & Antunes, C.H. (2006). Robustness Analysis in Multi-Objective Optimization Using a Degree of Robustness Concept, *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 6778-6783, Vancouver, Canada, July 2006, IEEE

Chen, W.; Sahai, A.; Messac, A. & Sundararaj, G. (1999). Physical Programming for Robust Design, *Proceedings of 40th Structures, Structural Dynamics and Materials Conference*, St. Louis, USA, April 1999

Coello, C.; Veldhuizen, D. & Lamont, G. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, ISBN 0306467623, Norwell

Das, I. (1997). *Nonlinear Multicriteria Optimization and Robust Optimality*, Rice University, PhD Thesis, Houston

Deb K. (2001). *Multi-Objective Optimisation Using Evolutionary Algorithms*, Wiley, ISBN 0-471-87339-X, Chichester

Deb, K.; Pratap, A.; Agrawal, S. & Meyarivan, T. (2002). A Fast and Elitist Multi-Objective Genetic Algorithm: NSGAII, *IEE Transactions on Evolutionary Computation*, Vol. 6, No. 2 (April 2002) 182-197, ISBN 1089-778X.

Ded, K.; Thiele, L.,; Laumanns, M. & Zitzler E. (2002). Scalable Multi-Objective Optimization Test Problems, *IEEE Transactions on Evolutionary Computation*, Vol. 1, (May 2002) 825-830, ISBN 0-7803-7282-4

Deb, K. & Gupta, H. (2005). Searching for robust Pareto-optimal solutions in multi-objective optimization, *Proceedings of the Third International Conference on Evolutionary Multi-*

Criterion Optimization, pp. 150-164, ISBN 978-3-540-24983-2, Guanajuato, Mexico, January 2005, Springer, Berlin

Deb, K. & Gupta, H. (2006). Introducing Robustness in Multi-objective Optimization. *Evolutionary Computation*, Vol. 14, No. 4, (December 2006) 463-494, 1063-6560.

Du, X. & Chen, W. (1998). Towards a Better Understanding of Modelling Feasibility Robustness in Engineering Design, *Proceedings of DETC 99*, Las Vegas, USA, September 1999, ASM

Ferreira, J.C.; Fonseca, C.M. & Gaspar-Cunha, A. (2008) Methodology to Select Solutions for Multi-Objective Optimization Problems: Weight Stress Function Method, *Applied Intelligence*, Accepted for publication (2008)

Fonseca, C.; Fleming, P. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, *Proceedings of Fifth International Conference on Genetic Algorithms*, pp. 416-423, University of Illinois, July 1993, Morgan Kauffman, Urbana-Champaign

Fonseca, C. & Fleming, P. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms, part I: A unified formulation, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, No. 1 (1998) 26-37

Gaspar-Cunha, A.; Oliveira, P. & Covas, J. (1997). Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems, *Proceedings of Seventh Int. Conf. on Genetic Algorithms*, pp. 682-688, Michigan, USA.

Gaspar-Cunha, A. (2000). *Modelling and Optimisation of Single Screw Extrusion*, University of Minho, PhD Thesis, Guimarães, Portugal

Gaspar-Cunha, A. & Covas, J. (2004). RPSGAe - A Multiobjective Genetic Algorithm with Elitism: Application to Polymer Extrusion, In: *Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems*, Gandibleux, X.; Sevaux, M.; Sörensen, K.; T'kindt, V. (Eds.), 221-249, Springer, ISBN 3-540-20637-X, Berlin

Gaspar-Cunha, A. & Covas, J. (2005). Robustness using Multi-Objective Evolutionary Algorithms, *Proceedings of 10th Online World Conference in Soft Computing in Industrial Applications*, pp. 189-193, ISBN 3-540-29123-7 (http://www.cranfield.ac.uk/wsc10/), September 2005, Springer, Berlin

Gaspar-Cunha, A. & Covas, J. (2008). Robustness in Multi-Objective Optimization using Evolutionary Algorithms, *Computational Optimization and Applications*, Vol. 39, No. 1, (January 2008) 75-96, ISBN 0926-6003

Goldberg, D. & Richardson, J. (1987). Genetic Algorithms with Sharing for Multimodal Function Optimization, *Proceedings of Second Int. Conf. on Genetic Algorithms*, pp. 41-49, 0-8058-0158-8, Cambridge, July 1985, Lawrence Erlbaum Associates, Mahwah

Goldberg, D. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, 0201157675, Reading

Guanawan, S. & Azarm, S. (2005). Multi-Objective Robust Optimization using a Sensitivity Region Concept, *Struct. Multidisciplinar Optimization*, Vol. 29, No. 1 (2005) 50-60

Horn, J.; Nafpliotis, N. & Goldberg, D. (1994), A Niched Pareto Genetic Algorithm for Multiobjective Optimization, *Proceedings of First IEEE Conference on Evolutionary Computation*, pp. 82-87, Jun 1994.

Jin, Y. & Sendhoff, B. (2003). Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach, *Proceedings of Second Int. Conf. on Evol.*

Multi-Objective Optimization, pp. 237-251, ISBN 3540018697, Faro, Portugal, April 2003, Springer

Jin, Y. & Branke, J. (2005). Evolutionary Optimization in Uncertain Environments – A Survey, *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 3, (June 2005) 303-317, 1089-778X

Kazancioglu, E.; Wu, G.; Ko, J.; Bohac, S.; Filipi, Z.; Hu, S.; Assanis, D. & Saitou, K. (2003). Robust Optimization of an Automobile Valvetrain using a Multiobjective Genetic Algorithm, *Proceedings of DETC'03*, pp. 1-12, Chicago, USA, September 2003, ASME

Knowles, J. & Corne, D. (2000). Approximating the Non-dominated Front using the Pareto Archived Evolutionary Strategy, *Evolutionary Computation*, Vol. 8, No. 2, (June 2000) 149-172, 1063-6560

Moshaiov, A. & Avigrad, G. (2006). Concept-Based IEC for Multi-Objective Search with Robustness to Human Preference Uncertainty, *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 6785-6791, Vancouver, Canada, July 2006, IEEE

Olvander, J. (2005). Robustness Considerations in Multi-Objective Optimal Design, *J. of Engineering Design*, Vol. 16, No. 5 (October 2005) 511-523

Paenk I., Branke, J. & Jin, Y. (2006). Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation, *IEEE Transations on Evolutionary Computation*, Vol. 10, No. 4 (August 2006) 405-420

Kouvelis, P. & Sayin, S. (2002). Algorithm Robust for the Bicriteria Discrete Optimization Problem, *Annals of Operational Research*, Vol. 147, No. 1, (October 2006) 71–85

Ray, T. (2002). Constrained Robust Optimal Design using a Multiobjective Evolutionary Algorithm, *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, pp. 419-424, ISBN 0-7803-7282-4, Honolulu, May 2002, IEEE

Ribeiro, J.L. & Elsayed, E.A. (1995). A case Study on Process Optimization using the Gradient Loss Funstion, *Int. J. Prod. Res.*, Vol. 33, No. 12, 3233-3248

Roseman, M. & Gero, J. (1985). Reducing the Pareto Optimal Set in Multicriteria Optimization, *Engineering Optimization*, Vol. 8, No. 3, 189-206, 0305-215X

Schaffer, J. (1984). *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*, Vanderbilt University, Ph. D. Thesis, Nashville

Sörensen, K. (2004). Finding Robust Solutions Using Local Search, *J. of Mathematical Modelling and Algorithms*, Vol. 3, No. 1, 89-103

Srinivas, N. & Deb, K. (1995). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, Vol. 2, No.3, 221-248

Tsutsui, S. & Ghosh, A. (1997). Genetic Algorithms with a Robust Solution Scheme, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 3, (September 1997) 201-208, 1089-778X

Wiesmann, D.; Hammel, U. & Bäck, T. (1998). Robust Design of Multilayer Optical Coatings by Means of Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 4, (November 1998) 162-167, 4235.738986

Zitzler, E.; Deb K. & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, Vol. 8, No. 2, (June 2000) 173-195, 1063-6560.

Zitzler, E.; Laumanns, M. & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK report, No. 103, Swiss Federal Institute of Technology, Zurich, Switzerland.
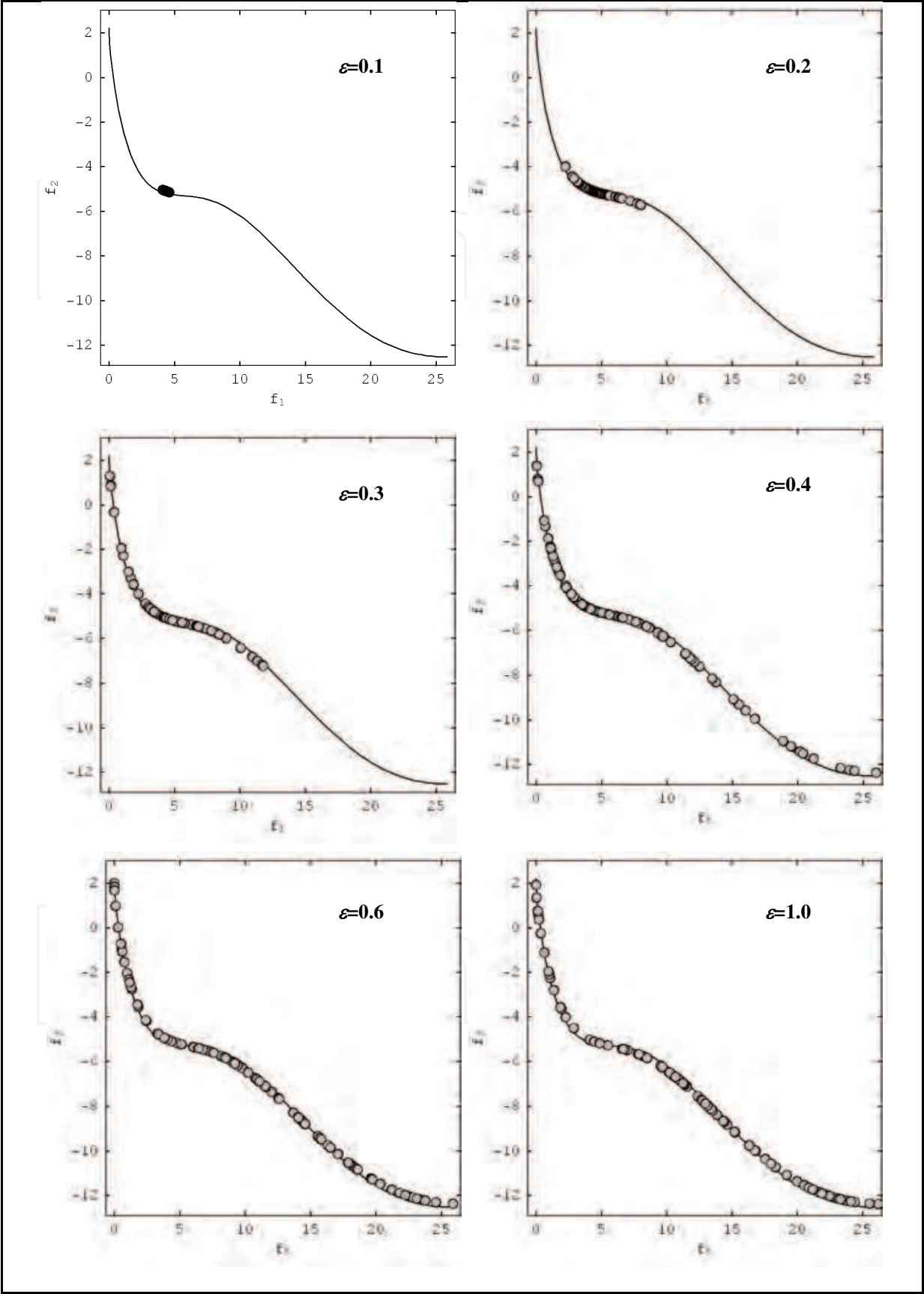
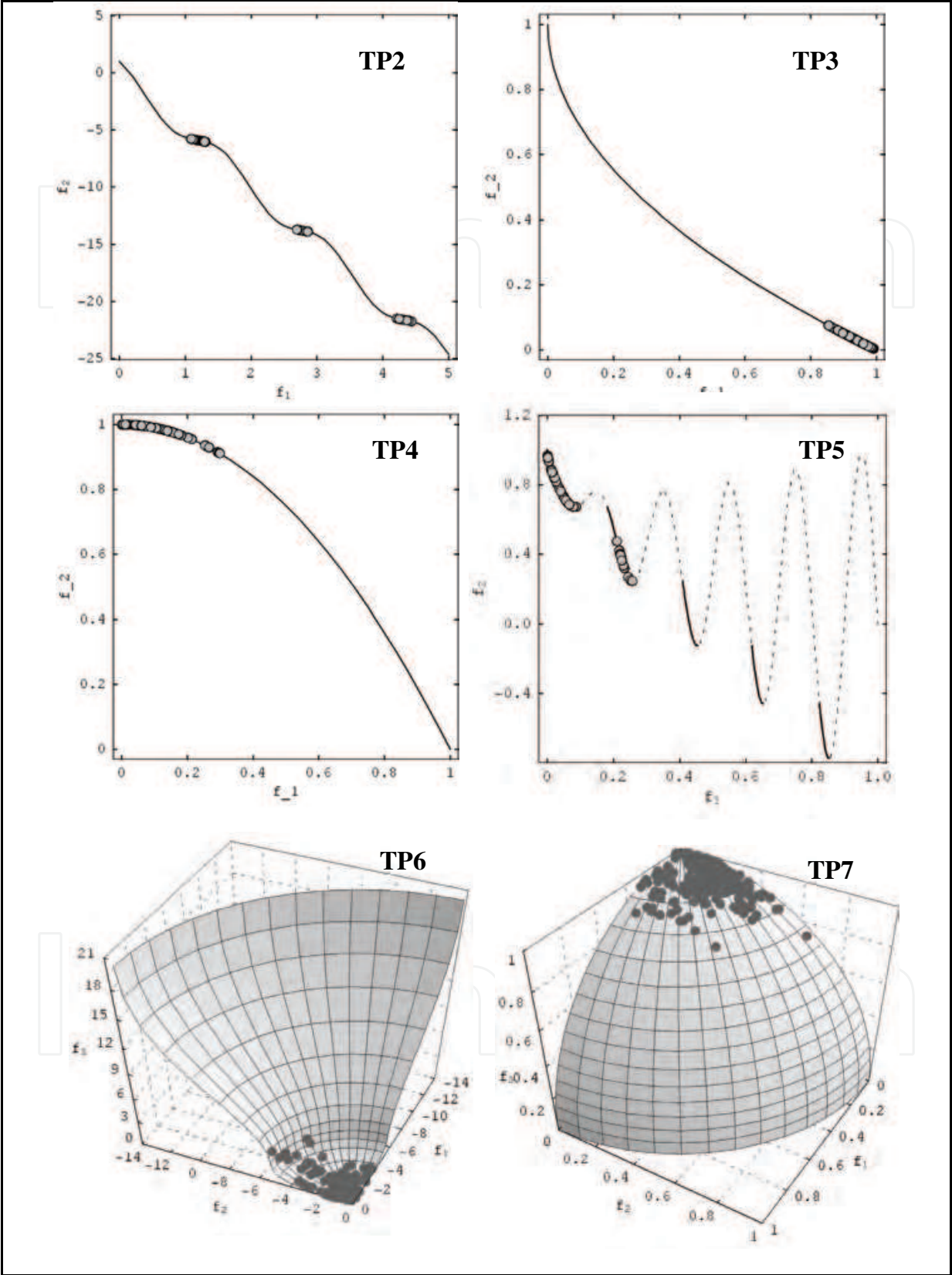Fig. 11. Influence of dispersion parameter for TP1

Fig. 12. Results for TP2 to TP7 (ε=0.1)

Advances in Evolutionary Algorithms
Edited by Xiong Zhihui

With the recent trends towards massive data sets and significant computational power, combined with evolutionary algorithmic advances evolutionary computation is becoming much more relevant to practice. Aim of the book is to present recent improvements, innovative ideas and concepts in a part of a huge EA field.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

J. Ferreira, C. M. Fonseca, J. A. Covas and A. Gaspar-Cunha (2008). Evolutionary Multi-Objective Robust Optimization, Advances in Evolutionary Algorithms, Xiong Zhihui (Ed.), ISBN: 978-953-7619-11-4, InTech, Available from: http://www.intechopen.com/books/advances_in_evolutionary_algorithms/evolutionary_multi-objective_robust_optimization