2$^{\text{nd}}$ International Conference on Engineering Optimization
September 6-9, 2010, Lisbon, Portugal

Assessment of an hybrid multi-objective pattern search filter method

M. Fernanda P. Costa[1] **and A. Gaspar-Cunha**[2]

[1] Department of Mathematics and Applications, University of Minho, Guimarães, Portugal,
mfc@math.uminho.pt
[2] IPC/I3N-Institute of Polymers and Composites, University of Minho, Guimarães, Portugal,
agc@dep.uminho.pt

Abstract
A hybrid multi-objective evolutionary algorithm (MOEA) for solving nonlinear multi-objective optimization problems that relies on a pattern search filter method is proposed. The aim is to reduce the computational time involved in solving expensive multi-objective problems by improving a subset of Pareto points. The proposed pattern search filter method relies on two components. Each entry in the filter aims to measure feasibility and optimality. The feasibility and optimality come directly from each single-objective nonlinear program problem that is associated to the multi-objective problem. Experiments carried out with a set of nonlinear multi-objective problems show that our pattern search filter approach is effective in reaching improved Pareto points. A comparison with other techniques known in the literature is presented.

Keywords: Multi-Objective, Nonlinear Programming, Pattern Search Method, Filter Method.

1. Introduction

Multi-objective optimization problems (MOOPs) arise in engineering and economic applications with multiple competing objectives. These applications often requires the use of complex "black-box" modeling routines able to link the design variables with the optimization criteria. This is frequently done through the resolution of the process governing equations and using expensive numerical methods, such as, finite differences, finite elements and finite volume elements [1, 2, 3, 4, 5, 6].

It is known that multi-objective evolutionary algorithms (MOEAs) are an excellent tool for solving this type of problems, where the problem functions are available as oracles, and derivative information is prohibited to obtain [7, 8]. MOEAs uses a population of potential solutions that evolves during successive generations by the application of genetic operators, such as selection, recombination, crossover and mutation [9]. In this technique the concept of non-dominance is used and a trade-off between the solutions (the Pareto frontier) is obtained [7]. Although MOAEs are good global search methods, due to their characteristics, there are some problems occurring in engineering area that require a large number of objective functions evaluations to obtain an acceptable solution - typically of the order of several thousands. Since this kind of problems appears very frequently in industrial process, it is crucial to reduce the number of objective functions evaluations to minimize the computational time involved in solving these expensive MOOPs [10]. A variety of proposals to reduce the number of objective functions evaluations have been reported in the literature, see for example [10, 11, 12, 13, 14, 15, 16, 18].

This paper aims to develop a hybrid method that combines a MOEA with a local search algorithm. The goal will be to accelerate the global search of the MOEA through the generation of trial solutions, using a local search algorithm based on a pattern search filter method (PSFM). More specifically, the PSFM will perform a local search in the neighborhood of some solutions selected from the Pareto frontier. A comparison between this hybrid method and two others MOEAs is presented.

2. Multi-Objective Optimization

The multi-objective optimization problem is formally defined as

$$\begin{array}{rrcl} \text{minimize}_{x\in\mathbb{S}} & f(x) & & \\ \text{subject to} & g(x) & = & 0 \\ & h(x) & \leq & 0 \end{array} \tag{1}$$

where $f(x) = (f_1(x), \ldots, f_M(x)) : \mathbb{R}^N \to \mathbb{R}^M$ are the M objective functions, $g(x) = (g_1(x), \ldots, g_J(x)) : \mathbb{R}^N \to \mathbb{R}^J$ and $h(x) = (h_1(x), \ldots, h_K(x)) : \mathbb{R}^N \to \mathbb{R}^K$ are the J equality and K inequality constraints, respectively. The search space $\mathbb{S}$ is defined as $\{x \in \mathbb{R}^N : x_i \in [l_i, u_i], i = 1, \ldots, n\}$. The feasible region, defined by the constraints, is denoted by $\overline{F} = \{x \in \mathbb{S} : g(x) = 0, h(x) \leq 0\}$, and we assume that is nonempty.

Pareto optimal and Pareto dominance are the most common concept of optimality in multiobjective optimization. Multiobjective optimization seeks to optimize the vector of objective functions over a feasible region in the space of decision variables. For the problem Eq.(1), the Pareto dominance and Pareto optimal are defined as follows.

Definition 1.(Pareto Dominance)
Given x^1, $x^2 \in \overline{F}$, the point x^1 is said to dominate the point x^2, if $f_m(x^1) \leq f_m(x^2)$ for all $m = 1, \ldots, M$, and there exists at least one $r \in \{1, \ldots, M\}$ such that $f_r(x^1) < f_r(x^2)$.

Definition 2. (Pareto Optimal)
Let $x^* \in \overline{F}$ be a feasible point with the corresponding criterion vector $z^* = f(x^*)$.

1. (x^*, z^*) is *globally Pareto-optimal* if there is no vector $x \in \overline{F}$, $x \neq x^*$, with $f_m(x) \leq f_m(x^*)$, for all $m = 1, \ldots, M$, and $f_r(x) < f_r(x^*)$ for at least one $r \in \{1, \ldots, M\}$.

2. (x^*, z^*) is *locally Pareto-optimal* if there exists a $\delta > 0$ such that $x^* \in \overline{F}$ is globally Pareto-optimal in $\overline{F} \cap B(x^*, \delta)$, where $B(x^*, \delta)$ is ball of radius δ around x^*.

We designate the set of all Pareto points as $\mathcal{P} = \{z^* : (x^*, z^*) \text{ is a Pareto point}\}$. The optimal solution of a MOOP is not a single optimal solution but a set of optimal solutions given by all the potential feasible solutions such that the multiple objective functions cannot be simultaneously improved. In practice it is impossible to know the actual optimal set and, consequently, the corresponding Pareto optimal front. Usually, the optimization algorithms find an approximation to this set.

2.1. Multi-Objective Evolutionary Algorithm

Evolutionary algoritms (EAs) are well recognized by its capacity to explore and find Pareto-optimal fronts for MOOPs [7]. For most problems MOEA outperforms traditional deterministic methods due to its capacity to explore and combine several solutions to find the Pareto front in a single run. In MOEA it is desirable to have an homogeneous distribution of the population along the Pareto frontier simultaneously with an improvement of the solutions along successive generations [2, 7, 17]. Generally MOEAs replace the selection phase of a traditional EA by a routine able to deal with multiple objectives.

The MOEA adopted in this work is the Reduced Pareto Set Genetic Algorithm (RPSGA) [8]. The main steps of this algorithm are shown in algorithm 1. Detailed of this algorithm can be found for instance in [3, 8]. The influence of some important parameters of the algorithm, such as, size of internal and external populations, number of individuals copied to the external population in each generation and from the external to the internal population and the limits of the indifference of the clustering technique, have already been studied. Thus, the values of the parameters inside the RPSGA are the best values described in [8].

Algorithm 1

```
0. counterG=0 (inner counter of generations).
1. Random initial population (internal).
2. Empty external population.
3. While not Stop-Condition do
  a) Evaluate internal population.
  b) Calculate the Fitness of the individuals using clustering.
  c) Copy the best individuals to the external population.
  d) If the external population becomes full then
       - Apply the clustering to this population.
       - Copy the best individuals to the internal population.
  e) Select the individuals for reproduction.
  f) Crossover.
  g) Mutation.
  h) Archive the best solutions.
  i) counterG=counterG+1.
```

3. Pattern Search Filter Method

In this section, we present a hybrid method that combines pattern search method with a filter method. Pattern search methods constitute a subclass of direct search methods. They are based on the generation of search directions that span the search space positively. In pattern search methods, exploratory moves from the current solution to trial points are made along pattern directions with a certain step size, to minimize a function with non-increasing objective function values. Thus, at each iteration the objective function is evaluated in a finite set of trial points, in order to find a new point yielding better objective function values. If these exploraty moves give no improvement, then the step size is decreased to refine the search. A review of pattern search methods can be found in [19].

The central notion in pattern search methods is the positive spanning. If $\{d_1, d_2, \dots, d_N\}$ is a basis in $\mathbb{R}^N$, two simple and common examples of positive basis are:

$$\begin{array}{l} D_+ = \{d_1, d_2, \dots, d_N, -\sum_{i=1}^N d_i\} \text{ and} \\ D_+ = \{d_1, d_2, \dots, d_N, -d_1, -d_2, \dots, -d_N\}. \end{array}$$

One of the simplest positive spanning sets is $D_\oplus = \{e_1, e_2, \dots, e_N, -e_1, -e_2, \dots, -e_N\}$, where $e_i \in \mathbb{R}^N$ is the ith unit vector in $\mathbb{R}^N$.

One technique to find a single Pareto point is to minimize one objective to achieving a given goal on all other objectives. Thus, in order to incorporate a local search based on a filter technique in the RPSGA, problem Eq.(1) is reformulated into M single-objective nonlinear program (NLP) of the form

$$\begin{array}{rrcl} \text{minimize}_{x\in\mathbb{S}} & f_m(x) & & \\ \text{subject to} & g(x) & = & 0 \\ & h(x) & \leq & 0 \\ & f_j(x) & \leq & f_j^{max_k}, \quad j = 1, \dots, M \,\text{and}\, j \neq m \end{array} \tag{2}$$

where $f_j^{max_k}$ is the maximum value of the objective function f_j at iteration k, and $m = 1, \dots, M$. For notational simplicity, the NLP (2) can be rewritten as follows:

$$\begin{array}{rrcl} \text{minimize}_{x\in\mathbb{S}} & f_m(x) & & \\ \text{subject to} & g(x) & = & 0 \\ & c(x) & \leq & 0 \end{array} \tag{3}$$

where the inequality constraints $c : \mathbb{R}^N \to \mathbb{R}^{K+M-1}$ are functions $c = (h_1, \dots, h_K, f_j(x) - f_j^{max_k}, \dots)$ for $j = 1, \dots, M$ and $j \neq m$.

The filter methods were originally proposed by Flecther and Leyffer [20], as an alternative to merit functions, to guarantee global convergence in algorithms for nonlinear programming. In the filter method the main idea is to interpret the NLP Eq.(3) as a bi-objective optimization problem with two goals: the minimization of a measure of the constrained violation $\theta_{f_m}(x) = \left\|(g(x), c(x)^+)^T\right\|$ (with $\|a^+\| = \|max(a, 0)\|$ for some norm) and the minimization of the objective function $f_m(x)$.

The filter technique attempts to minimize both functions, but a certain emphasis is placed on the first objective, θ_{f_m}, since convergence to a feasible point must be ensured. Filter methods are based on the concept of dominance, borrowed from multiobjective optimization. The filter $\mathcal{F}_{f_m}$ is defined as a finite set of pairs $(\theta_{f_m}(x_l), f_m(x_l))$, that correspond to a collection of previous iterates $x_l \in \mathbb{S}$, with the additional requirement that no filter entry is dominated by any of the others.

The filter method proposed in this paper, to improve a subset of points $S = \{x_k^1, \dots, x_k^s\}$ selected from the Pareto frontier at iteration k, will deal with problem Eq.(1) through the M reformulated problems of the form Eq.(3).

Therefore, for each $x_k^i \in S$ (with $i = 1, \dots, s$) the PSFM method employs a backtracking strategy generating a decreasing sequence of step-length control parameters

$$\alpha_{i,l} \in (0, \alpha_0], l = 0, 1, \dots,$$

until a set of acceptance conditions are satisfied. In this approach, a trial point $x_k^i(\alpha_{i,l}) = x_k^i + \alpha_{i,l} d$, for $d \in D_+$, might be acceptable by the filter, if it leads to sufficient progress in one of the two measures compared to the current iterate,

$$\theta_{f_m}(x_k^i(\alpha_{i,l})) \leq \ (1 - \gamma_\theta)\theta_{f_m}(x_k^i) \ \text{ or } \ f_m(x_k^i(\alpha_{i,l})) \leq f_m(x_k^i) - \gamma_f \theta_{f_m}(x_k^i). \tag{4}$$

However, to prevent convergence to a feasible but nonoptimal point, the above criteria are replaced by requiring only sufficient progress in the objective function, whenever the current iterate has $\theta_{f_m}(x_k^i) \leq \theta_{f_m}^{min}$, for some positive constant $\theta_{f_m}^{min}$. Thus, if $\theta_{f_m}(x_k^i) \leq \theta_{f_m}^{min}$ is true, the trial point has to satisfy:

$$f_m(x_k^i(\alpha_{i,l})) \leq f_m(x_k^i) - \gamma_f \theta_{f_m}(x_k^i), \tag{5}$$

in order to be acceptable. Iteration l will be designated a f-type iteration if the trial point satisfies this condition. Besides requiring sufficient decrease with respect to the current iterate, the trial point $x_k^i(\alpha_{i,l})$ is accepted only if it is acceptable by the current filter.

At the beginning of each optimization process, the filter is initialized as follows:

$$\mathcal{F}_{f_m}^1 = \{(\theta_{f_m}, f_m) \in \mathbb{R}^2 : \theta_{f_m} \geq \theta_{f_m}^{\max}\} \tag{6}$$

for some nonnegative constant $\theta_{f_m}^{\max}$, and is updated whenever the accepted trial point satisfies condition (4) by

$$\mathcal{F}_{f_m}^{i+1} = \mathcal{F}_{f_m}^i \cup \{(\theta_{f_m}, f_m) \in \mathbb{R}^2 : \theta_{f_m} \geq (1-\gamma_\theta)\theta_{f_m}(x_k^i),\ f_m \geq f_m(x_k^i) - \gamma_f \theta_{f_m}(x_k^i)\}. \tag{7}$$

The filter remains unchanged if (5) holds for the accepted trial point. Algorithm 2 describes the steps of the PSFM used to improve a set of points selected from the Pareto frontier through of the one of the M reformulated problems Eq.(3).

```
Algorithm 2
1. Initialization
Let $\alpha_{tol} > 0$ be the stopping tolerance.
Let $\alpha_0 > \alpha_{tol}$ be the initial value of the step length control parameter.
Let $D = D_+$ be the positive spanning set.
Let $\{x_k^1, \ldots, x_k^{N_{sol}}\}$ be the $N_{sol}$ select points from the Pareto frontier at iteration $k$.
2. Search Step
For $i = 1, \ldots, N_{sol}$ do
  a. If $i == 1$ then, initialize the filter using (6). Set $l = 0$ and set $\alpha_{i,l} = \alpha_0$.
  b. If there exists $d \in D$ such that $(\theta_{f_m}(x_k^i(\alpha_{i,l})), f_m(x_k^i(\alpha_{i,l}))) \notin \mathcal{F}_{f_m}$, with $x_k^i(\alpha_{i,l}) = x_k^i + \alpha_{i,l} d$, and
     ((5) or (4)) hold, then:
       – Set $x_{k+1}^i = x_k^i(\alpha_{i,l})$ (accept a new iterate).
       – If $l$ is not a $f$-type iteration, increase the filter using (7). Otherwise, leave the filter unchanged.
  c. Otherwise, $(\theta_{f_m}(x_k^i(\alpha_{k,l})), f_m(x_k^i(\alpha_{k,l}))) \in \mathcal{F}_{f_m}$, for all $d \in D$, or ((5) and (4)) does not hold, do
     the following:
       – Reject the trial point $x_k^i(\alpha_{i,l})$.
       – Set $\alpha_{i,l+1} = \frac{1}{2}\alpha_{i,l}$, (contract the step-length control parameter).
       – Set $l = l + 1$.
       – If $\alpha_{i,l} < \alpha_{tol}$, then
           - Set $x_{k+1}^i = x_k^i$ (i.e., if not found a point acceptable by the filter)
           - Terminate.
```

In our implementation, the square of the Euclidean norm is used to measure the infeasibility, $\theta_{f_m}(x)$. The default values for PSFM are: $\theta_{f_m}^{\max} = 10^4 \max_{i=1,\ldots,N_{sol}}\{1, \theta_{f_m}(x_k^i)\}$, $\theta_{f_m}^{min} = 10^{-6}$, $\gamma_\theta = \gamma_f = 10^{-5}$, $\alpha_{tol} = 10^{-6}$, $\alpha_0 = 1.0$. The positive spanning set is defined as $D_+ = \{\beta_1 e_1, \ldots, \beta_N e_N, -\beta_1 e_1, \ldots, -\beta_N e_N\}$, where $\beta_j = \max(x_{k_j}^i - l_j^{min_k}, u_j^{max_k} - x_{k_j}^i)/e$, where $e \in \{2, 5, 10\}$, for $j = 1, \ldots, N$. u^{max_k} and l^{min_k} are the maximum and the minimum values of the decision variables at iteration k, respectively. The PSFM local search procedure is applied at every N_{gen} generations. The PSFM approach depends on three parameters: N_{gen}, N_{sol} and e. The influence of all these parameters will be analyzed.

4. The Hybrid Multi-Objective Pattern Search Filter Method

The RPSGA approach is combined with the PSFM to form the hybrid method RPSGA&PSFM, which is expected to have a higher ability to improve a subset of Pareto points. In the RPSGA&PSFM, we first run the RPSGA a predefined number of generations, N_{gen}. In each N_{gen} generations, a set of non-dominated solutions is selected from the current Pareto frontier and the PSFM is applied to improve this set. Specifically, we apply the PSFM M times, one per each (NLP) Eq.(3), and M new sets of the improved solutions are obtained. The hybrid algorithm RPSGA&PSFM is implemented by incorporating the following steps (Algorithm 3) between the selection and crossover phases in Algorithm 1.

```
Algorithm 3
If counterG = $N_{gen}$ then
    - Select the $N_{sol}$ solutions using the clustering technique: $S = \{x_k^1, \ldots, x_k^{N_{sol}}\}$
    - Set $S_{new} = \{\}$
    - For $m = 1, \ldots, M$ do
        - Considering the NLP (3), respective.
        - Apply the PSFM to get new set of solutions, $S_{f_m}$, by improving the set $S$.
        - Set $S_{new} = S_{new} \cup S_{f_m}$
Incorporate $S_{new}$, the new solutions generated, in the main population.
Set counterG = 0.
```

5. Experimental Results and Conclusions

The RPSGA&PSFM approach proposed will be tested using ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 benchmark problems (see Table in the appendix at the end of the paper). These functions cover various types of Pareto-optimal fronts, such as convex (ZDT1), non-convex (ZDT2), discrete (ZDT3), multimodal (ZDT4) and non-uniform (ZDT6) [7]. The performance of the methods was assessed using the hypervolume metric [21]. Due to the difficulty in comparing Pareto sets there is no consensus on the literature about the best metric to be used [7, 21]. However, take into account that, in this case, the aim is to check the evolution of the algorithm performance along the successive generations, the hypervolume metric seems to be a good choice. Hypervolume will be analyzed against the number of objective functions evaluations, since the goal is to reduce this number in order to get, simultaneously, identical performance. Due to the stochastic nature of the initial trial solutions it was necessary to perform several runs for each case study. Thus, an average of hypervolume metric is presented.

Two studies were performed. First, using all test problems, the influence of the algorithm parameters on the performance was studied. Then, a comparison between RPSGA, RPSGA&PSFM and NSGA-II algorithms was performed considering the best parameters defined previously. We remark that, the NSGA-II algorithm is one the most popular MOEAs [17]. Thirty runs for each case have been performed using the following RPSGA parameters (see [8] for more details): the main and elitist populations had 100 and 200 individuals, respectively; a roulette wheel selection strategy was adopted; a crossover probability of 0.8, a mutation probability of 0.05, a number of ranks of 30 and limits of indifference of the clustering technique of 0.01 were chosen.

Since, in the test problems selected the objectives are to be minimized and the algorithm is prepared to maximize, the following transformation into the original objective function (f) is used:

$$f^*(x) = \frac{1}{f(x)+1}. \tag{8}$$

In this case the function values were normalized in the interval [0,1]. The hypervolume metric was calculated using the new objective function search space defined by this transformation. Figures 1-3 show the evolution of the hypervolume metric (average of 30 runs with different seed values) with the number of evaluations, for the ZDT1 test problem, in function of the parameters: N_{sol}, N_{gen} and e. Identical study was made for each one of the test problems.

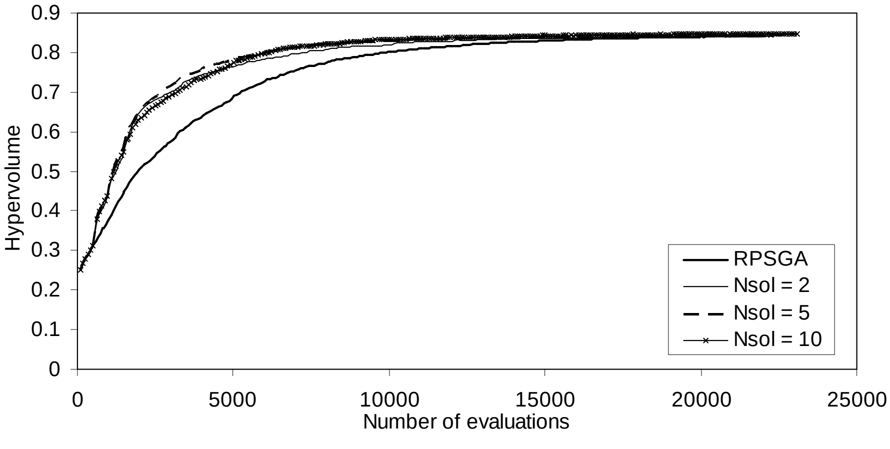

Figure 1: Influence of the N_{sol} parameter

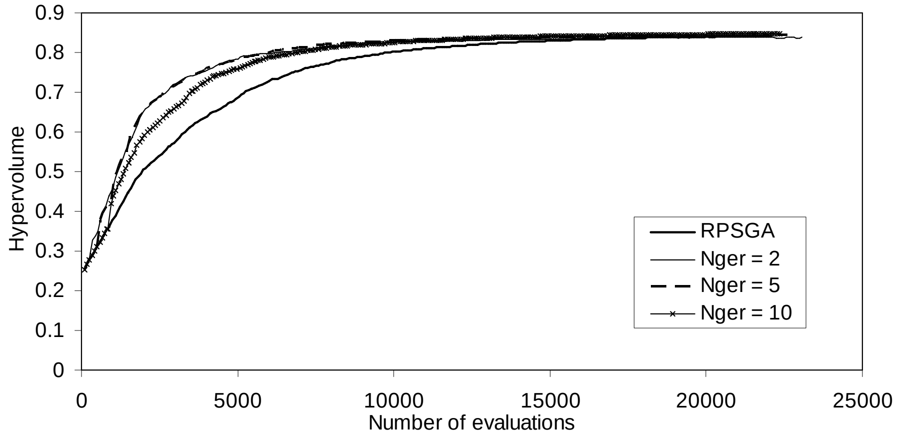

Figure 2: Influence of the N_{gen} parameter

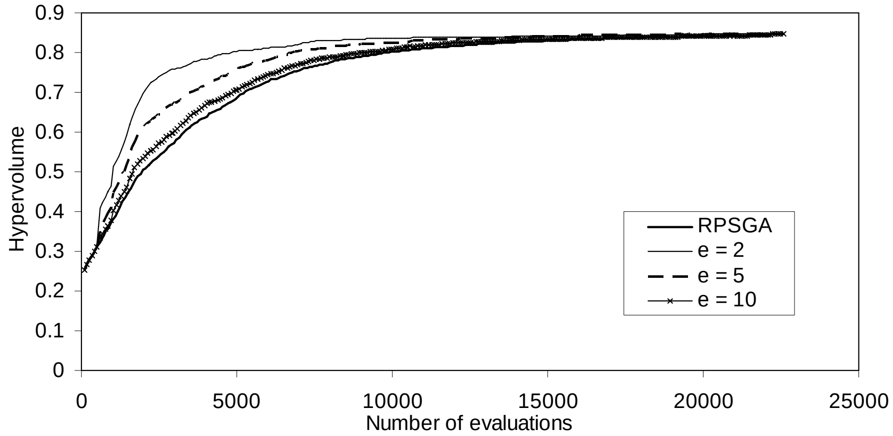

Figure 3: Influence of the N_{gen} parameter

As it can be seen in Figures 1-3 the convergence of the RPSGA&PSFM proposed is much higher than RPSGA alone. For the ZDT1 test problem, the best values of the parameters are: $N_{sol} = 5$, $N_{gen} = 5$ and $e = 2$. For the remaining test problems the best parameters are: $N_{sol} = 10$, $N_{gen} = 10$ and $e = 2$ for ZDT2 and ZDT3; $N_{sol} = 10$, $N_{gen} = 5$ and $e = 2$ for ZDT4; and $N_{sol} = 2$, $N_{gen} = 2$ and $e = 10$ for ZDT6. Thus, these are the values adopted in the remaining runs below.

Figures 4-8 compare the best results obtained by the RPSGA alone, the RPSGA&PSFM and the NSGA-II approaches. As it can be seen, the most important characteristics of RPSGA&PSFM is the fast improvements during the initial generations for all test problems. More specifically, the RPSGA&PSFM method is able to impose high pressure on the search at the beginning of the process, when the distance to the optimum region is high. Thus, it is clear, from these results, that the application of local search is strongly relevant at the initial stages of the search. However, this can lead the search to local regions of the search space. This bias is avoided by a careful balance between the solutions generated by the PSFM and those generated by the RPSGA. Clearly the hybrid method RPSGA&PSFM outperfoms the other methods in most of the test problems tested except in the case of the ZDT3.

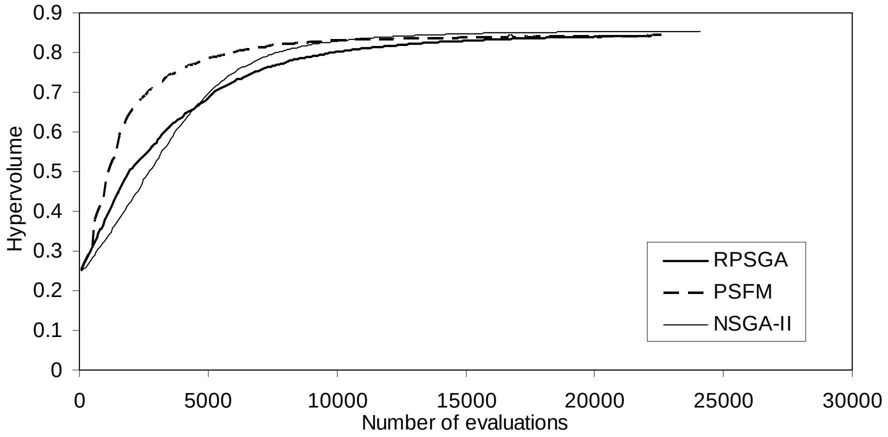

Figure 4: Hypervolume versus number of evaluations for ZDT1 test problem

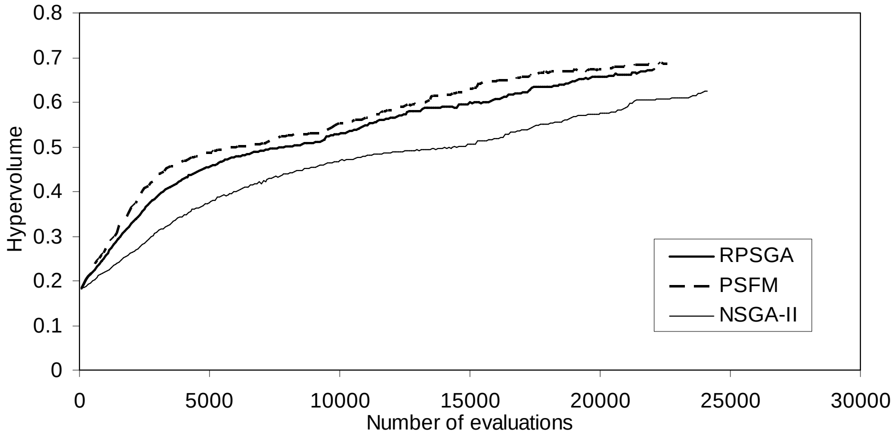

Figure 5: Hypervolume versus number of evaluations for ZDT2 test problem

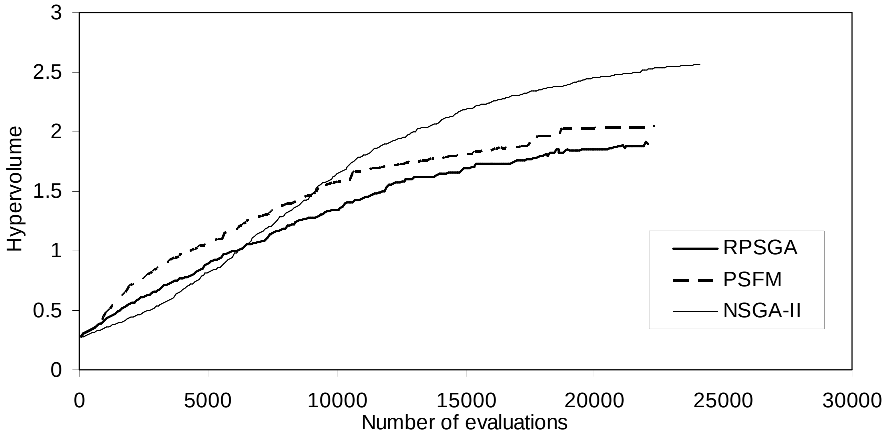

Figure 6: Hypervolume versus number of evaluations for ZDT3 test problem.

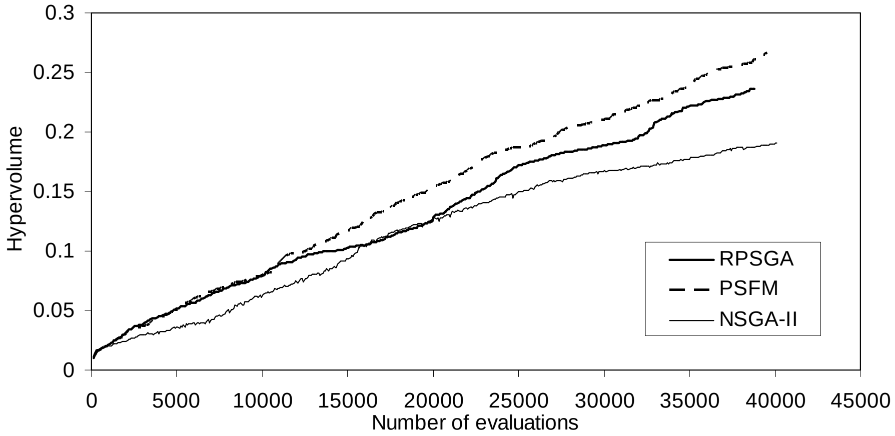

Figure 7: Hypervolume versus number of evaluations for ZDT4 test problem.

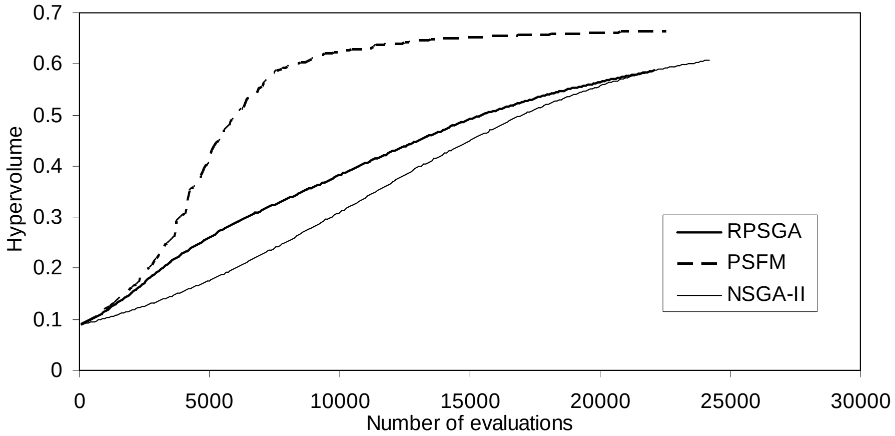

Figure 8: Hypervolume versus number of evaluations for ZDT6 test problems.

Acknowledgements
This work was financially supported by the Research Centre of Mathematics of the University of Minho through the FCT Pluriannual Funding Program.

References

[1] O.S. Carneiro and J.M. Nobrega, Recent developments in automatic die design for profile extrusion, *Plastics Rubber and Composites*, 33, 400-408, 2004.

[2] J.A. Covas, A. Gaspar-Cunha and P. Oliveira, An Optimization Approach to Practical Problems in Plasticating Single Screw Extrusion, *Polym. Eng. and Sci.*, 39 (3), 443-456, 1999.

[3] A. Gaspar-Cunha, Modeling and Optimization of Single Screw Extrusion, *PhD. Thesis*, University of Minho, Portugal, 2000.

[4] A. Gaspar-Cunha, A. Poulesquen, B. Vergnes and J.A. Covas, Optimization of Processing Conditions for Polymer Twin-Screw Extrusion, *Intern. Polym. Process.*, 17(3), 201-213, 2002.

[5] J.M. Nobrega and O.S. Carneiro, Optimising cooling performance of calibrators for extruded profiles, *Plastics Rubber and Composites,* 35, 387-392, 2006.

[6] C. Rauwendaal, Polymer Extrusion, *Hanser Publishers*, Munich, 1986.

[**7**] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, Chichester, 2001.

[**8**] A. Gaspar-Cunha and J.A. Covas, RPSGAe-A Multiobjective Genetic Algorithm with Elitism: Application to Polymer Extrusion, Metaheuristics for Multiobjective Optimisation, *Lecture Notes in Economics and Mathematical Systems*, Springer, 2004.

[**9**] D.E. Goldberg, Genetic Algorithms in Search, *Optimisation and Machine Learning*, Addison-Wesley, 1989.

[**10**] Y. Jin, M. Olhofer and B. Sendhof, A Framework for Evolutionary Optimization with Approximate Fitness Functions, *IEEE Trans. on Evolutionary Computations*, 6, 481-494, 2002.

[**11**] A. Gaspar-Cunha and A. Vieira, A Multi-Objective Evolutionary Algorithm Using Approximate Fitness Evaluation, *International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, EUROGEN, CIMNE, 2003.

[**12**] A. Gaspar-Cunha, A.S. Vieira and C.M. Fonseca, Multi-Objective Optimization: Hybridization of an Evolutionary Algorithm with Artificial Neural Networks for fast Convergence, *Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics*, 2004.

[**13**] A. Gaspar-Cunha and A. Vieira, A Multi-Objective Evolutionary Algorithm Using Neural Networks To Approximate Fitness Evaluations, *International Journal of Computers, Systems, and Signals*, 6, 18-36, 2005.

[**14**] P.K.S. Nain and K. Deb, A Computationally Effective Multi-Objective Search and Optimization Technique Using Coarse-to-Fine Grain Modelling, *Kangal Report No. 2002005*, Available via DIALOG urlhttp://www.iitk.ac.in/kangal/deb.htm, 2002.

[**15**] C. Poloni, A. Giurgevich, L. Onesti, and V. Pedirola, Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for complex design problem in fluid dynamics, *Computer Methods in Applied Mechanics and Engineering*, 186, 403-420, 2002.

[**16**] C. Poloni, A. Giurgevich, L. Onesti and V. Pedirola, Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for complex design problem in fluid dynamics, *Computer Methods in Applied Mechanics and Engineering*, 186, 403-420, 2000.

[**17**] K. Deb, S. Agrawal, T. Meyarivan and A. Pratap, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGAII, *Proceedings of the Parallel Problem Solving from Nature VI*, 849-858, 2000.

[**18**] A.K.A. Talukder, M. Kirley and R. Buyya, A pareto following variation operator for fast-converging multiobjective evolutionary algorithms, *Proceedings of the 10th annual conference on Genetic and evolutionary*, 721-728, 2008.

[**19**] T.G. Kolda, R.B. Lewis and V. Torczon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, 45(3), 385-482, 2003.

[**20**] R. Fletcher and S. Leyffer, Nonlinear Programming without a Penalty Function, *Mathematical Programming*, 91, 239-269, 2002.

[**21**] E. Zitzler, K. Deb and L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, 8, 173-195, 2000.

Appendix - Table of test problems

Name	N	Objective	x
ZDT1	30	$f_1(x) = x(1)$	$x \in [0,1]$
		$f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}}\right)$	
		where $g(x) = 1 + 9\,\frac{\sum_{i=2}^{N} x(i)}{N-1}$	
ZDT2	30	$f_1(x) = x(1)$	$x \in [0,1]$
		$f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$	
		where $g(x) = 1 + 9\,\frac{\sum_{i=2}^{N} x(i)}{N-1}$	
ZTD3	30	$f_1(x) = x(1)$	$x \in [0,1]$
		$f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)}\right)\sin\left(10\pi f_1(x)\right)\right)$	
		where $g(x) = 1 + 9\,\frac{\sum_{i=2}^{N} x(i)}{N-1}$	
ZTD4	10	$f_1(x) = x(1)$	$x(1) \in [0,1]$
		$f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}}\right)$	$x(i) \in [-5,5]$
			$(i = 2, \ldots, N)$
		where $g(x) = -9 + 10N + \sum_{i=2}^{N}\left(x(i)^{2} - 10\cos(4\pi x(i))\right)$	
ZDT6	10	$f_1(x) = 1 - \exp(-4x(1))\sin^6(6\pi x(1))$	$x \in [0,1]$
		$f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$	
		where $g(x) = 1 + 9\left(\frac{\sum_{i=2}^{N} x(i)}{N-1}\right)^{0.25}$	