g-dominance: Reference point based dominance for MultiObjective Metaheuristics

Julián Molina [c] Luis V. Santana [a] Alfredo G. Hernández-Díaz [b]
Carlos A. Coello Coello [a] Rafael Caballero [c]

[a] *CINVESTAV-IPN, Computer Science Department, México*
[b] *Pablo de Olavide University, Seville, Spain*
[c] *University of Málaga, Málaga, Spain*

Abstract

One of the main tools for including decision maker (DM) preferences in the multiobjective optimization (MO) literature is the use of reference points and achievement scalarizing functions [33]. The core idea in these approaches is converting the original MO problem into a single-objective optimization problem through the use of a scalarizing function based on a reference point. As a result, a single efficient point adapted to the DM's preferences is obtained. However, a single solution can be less interesting than an approximation of the efficient set around this area, as stated for example by Deb in [14]. In this paper, we propose a variation of the concept of Pareto dominance, called g-dominance, which is based on the information included in a reference point and designed to be used with any MO evolutionary method or any MO metaheuristic. This concept will let us approximate the efficient set around the area of the most preferred point without using any scalarizing function. On the other hand, we will show how it can be easily used with any MO evolutionary method or any MO metaheuristic (just changing the dominance concept) and, to exemplify its use, we will show some results with some state-of-the-art-methods and some test problems.

Key words: Multiple criteria decision making, interactive methods, preference information, reference point.

Email address: jumolina@uma.es (Julián Molina).

Draft version: *17 November 2007*

1 Introduction

Multiple criteria optimization naturally appears in most real-world applications, and the term MultiObjective Programming (**MOP**) problem refers to such problems. The first difficulty that we face when dealing with Multiobjective Optimization (MO) is that the notion of "optimum" changes. In this case, rather than aiming to find the *global optimum*, we look for good trade-offs among the objectives, which are obtained by using the definition of Pareto efficiency. Such a definition will lead us to obtain not one, but a set of (Pareto) efficient solutions (the Pareto front, PF).

The idea of solving a multiobjective optimization problem is understood as helping a human Decision Maker (DM) in considering the multiple criteria simultaneously and in finding a Pareto efficient solution that pleases him/her the most. More details about the resolution of a **MOP** can be found in [31].

The common element in all **MOP** techniques is the need to find a sufficiently wide and representative set of efficient points where the DM is able to find an alternative adjusted to his/her preferences. A commonly adopted approach to find this type of solutions are the so-called Interactive Multi-Objective methods (see Miettinen [25]), which assume that the DM is able to provide consistent feedback regarding which preferences to include in the resolution process. This interaction can guide a search towards the most preferred areas of the Pareto front obtained and avoids exploring non-interesting solutions. These methods are very useful in real-world cases, as they help the DM to find the most preferred solutions in a consistent and reliable way.

The main problem when solving a real application is that some of the existing methods generate the entire Pareto set (most of the MO metaheuristics) whilst others produce a single point (most of the Interactive Multi-Objective methods). Our aim in this paper is producing something in-between. Thus, we will show how the use of g-dominance within a MO metaheuristic will let us produce a (reduced) set of efficient points adapted to the DM's preferences instead of the entire Pateto Set or a single efficient solution.

One of the main tools for expressing preference information is the use of reference points [33]. Reference points consist of aspiration levels reflecting desirable values for the objective functions. This is a natural way of expressing preference information and lets the DM express hopes about his/her most preferred solutions. The reference point is projected onto the Pareto front by minimizing a so-called achievement scalarizing function [33] outlined in Section 3. Reference points and achievement scalarizing function play the main role in some of the most commonly adopted methods, such as the light beam search [19], the visual interactive approach [21] and the Pareto Race [22], the

STOM method [26] or the NIMBUS method [24].

When solving real-world optimization problems, classical methods encounter great difficulty in dealing with the complexity involved in these situations and cannot offer a reliable solution. We can find many real applications in fields such as economics, engineering or science where methods with ample mathematical support (ensuring the optimality of solutions under ideal conditions) cannot obtain a solution or cannot obtain one in a reasonable time. These facts led researchers to develop metaheuristic methods to solve these very complex models. The success of these types of strategies produced enormous interest in their study giving rise to an active community and a number of very efficient metaheuristic algorithms for multiobjective optimization. Such approaches, which are generically called MultiObjective Meta-Heuristics (MOMH) are very popular nowadays, as shown in several surveys such as [20], [15] or [5]. However, most of the MOMH focus on the approximation of the Pareto front without including DM's preferences. However, as shown before, the determination or approximation of the Pareto front is not enough, and the DM's preferences have to be incorporated in order to determine the solution that better represents these preferences. But very few works can be found using MOMH which incorporate DM's preferences (and will be shown in Section 2) and a common fact in all of them is that many modifications on the main architecture have to be done in order to include DM's preferences into the MOMH.

This is an important fact when dealing with a complex problem because not every MOMH can be suitable or efficient for any given problem. In the MOMH literature, one can find efficient MOMH for nonlinear continuous problems, for combinatorial problems, for problems where evaluating the objective functions is very expensive, for vehicle routing problems, and for many other types of complex problems. Thus, we can say that, regardless of the type of problem to be solved, one can find an efficient MOMH method to deal with it. However, in most cases, these MOMH methods are not designed to including DM's preferences and modifying it for such an aim may be cumbersome, as we will see when reviewing MOMH methods including preferences.

In this paper, we propose a new concept of dominance that will allow us to include easily the DM's preferences into any MOMH, without having to modify the main architecture of the specific search engine adopted. This concept will combine the traditional Pareto efficiency (that will be defined in Section 1.1) with the use of reference points (that will be described in Section 3), and will be designed to be used together with a MO metaheuristic in order to let it easily include the DM's preferences. As mentioned before, one of the main advantages with respect to the existing attempts to include preferences when using a MO metaheuristic (that will be described in Section 2) is that g-dominance can be used without having to modify the main architecture

of the main method, as will be shown in Section 4. Finally, in Section 5 we will validate our proposed approach implementing the g-dominance in two different metaheuristics: the NSGA-II [12], which is a MOEA representative of the state-of-the-art in the area, and the DEMORS method [29], which is a hybrid of a differential evolution method with a Rough Sets tool.

1.1 Pareto efficiency

Given the MultiObjective Programming problem (**MOP**):

$$\textbf{(MOP)}\ Min\ (\ f_1\ (\ \mathbf{x}\ )\ ,\ f_2\ (\ \mathbf{x}\ )\ ,\ ...\ ,\ f_p\ (\ \mathbf{x}\ )\ )$$

$$\text{s. t.} : \mathbf{x} \in X$$

where:

$\cdot \mathbf{x} = (\ x_1\ ,\ x_2\ ,\ ...\ ,\ x_n\ )$ are the decision variables,

$\cdot X$ is the set of feasible solutions.

$\cdot f_i$ are the objective functions.

$\cdot \mathbf{f} = (\ f_1\ ,\ f_2\ ,\ ...\ ,\ f_p\ )$ is called the vector objective function.

A feasible solution $\mathbf{x^*} \in X$ is (Pareto) efficient for the **MOP** problem if there does not exist any other solution $\mathbf{x} \in X$, such that:

$$f_i\ (\ \mathbf{x}\ ) \leq f_i\ (\ \mathbf{x^*}\ )\ \forall i = 1,\ ...\ ,\ p$$

with at least one $j \in \{\ 1,\ ...\ ,\ p\ \}$ such that $f_j(\mathbf{x}) < f_j(\mathbf{x^*})$.

If this is not the case, this solution $\mathbf{x^*}$ is said to be dominated by solution $\mathbf{x}$. The set of all the efficient solutions for **MOP** is called the Pareto front.

2 Including Preferences with a MultiObjective Metaheuristic

As indicated before, only a few works can be found using MOMH and including DM's preferences. In [3,4], we can find a survey on including preferences when using a multiobjective evolutionary algorithm (MOEA). The following are some of the methods reviewed therein:

In [16], we can find the earliest attempt to incorporate preferences, and the proposal was to use MOGA (introduced in the same paper) together with

goal information as an additional criterion to asign ranks to the population. Greenwood and Hu [17] adopt utility functions to perform ranking of attributes, and also incorporate preference information into the survival criteria. Cvetkovic and Parmee, [6] and [7], use binary preference relations (translated into weights) to narrow the search. These weights are used in different ways to modify the concept of dominance. Rekiek et al. [29] use the PROMETHEE method to generate weights for a MOEA. Massebeuf et al. [23] use PROMETHEE II in an *a posteriori* form: a MOEA generates efficient solutions and PROMETHEE II selects some of them based on the DM's preferences. In [8,12], Deb uses variations of Compromise Programming to bias the search of a MOEA. Finally, in [10,11], Deb requires the DM to provide specific goals for each objective.

More recently, some other approaches can be found, as in [27] where Phelps and Koksalan use pairwise comparisons to include the DM's preferences into the fitness funcion. In the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) proposed in [2], user preferences are taken into account using trade-offs, supplied by the DM, in order to modify the definition of dominance. In [1], Branke and Deb propose two schemes to include preference information when using a MOEA (they use the NSGA-II [13] for validation purposes): (1) modifying the definition of dominance (using the Guided Dominance Principle of G-MOEA) and (2) using a biased crowding distance based on weights. In Deb et. al [14], preferences are included through the use of reference points. In this paper, the authors claim that "*a single solution does not provide a good idea of the properties of solutions near the desired region of the front*" and that "*by providing a (reference point), the decision-maker is not usually looking for a single solution, rather she/he is interested in knowing the properties of solutions which correspond to the optimum and near-optimum solutions respecting the (reference point).*" But the approach followed to get this approximation is based on rankings and then can only be applied with ranking based methods, such as the NSGA-II. Also, being the case of a ranking-based method, important modifications of the main algorithm have to be carried out.

In [28] (using a tabu search and simulated annealing method) and in [32] (using a simulated annealing method) the authors ask the DM to provide levels for each of the objectives at each iteration, and use such levels to constrain the solution space to be explored. Finally, Hapke et al. [18] (using a simulated annealing method) compute the approximation of the Pareto front and then invoke an interactive procedure to find the most preferred solution within that set.

Summarizing, all of these methods require important modifications to the MOMH used as a search engine in order to generate the Pareto front, and then it becomes more difficult to introduce further modifications for incorporating the DM's preferences. In general, a change in the main architecture of

the MOMH is required for incorporating user's preferences. This makes things difficult for practicioners, who are normally interested only in a small set of efficient solutions rather than the entire Pareto front. Thus, to solve a MOP problem, one must be able to find efficient solutions (i.e., resolution capabilities) and must be able to interact with the DM (i.e., interaction capabilities) in order to incorporate his/her preferences during the search process. But, as shown in Figure 1, the most suitable method (Evolutionary Algorithms (EMO), Tabu Search (TS), Scatter Search (SS), etc) to solve a given problem can be very difficult to modify in order to incorporate interaction, and then one can be forced to change the MOMH adopted.

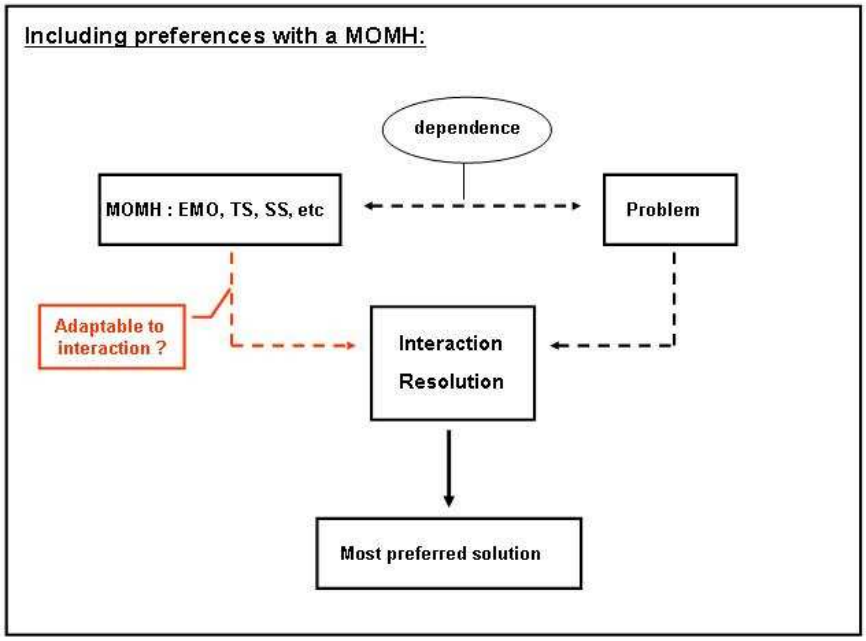

Fig. 1. Including DM's preferences

3 Reference points and Achievement Scalarizing Functions

Achievement scalarizing functions **(asf)** were first proposed in [33] and nowadays are part of many MOP methods. The achievement (scalarizing) function projects any given (feasible or infeasible) reference point $\mathbf{g} \in \mathbb{R}^p$ onto the Pareto front. Also, as shown in [25] any efficient solution can be found using an **asf**. This approach transforms a MultiObjective Optimization problem (**MOP**) into the following single-objective problem **(ASFP)**:

$$\textbf{(ASFP)} \quad Min\ s_{\mathbf{g}}(\mathbf{f}(\mathbf{x})) = \max_{i=1,\cdots,p} \{\omega_i(f_i(\mathbf{x}) - g_i)\} + \rho \sum_{i=1}^{p} (f_i(\mathbf{x}) - g_i)$$

$$\text{s. t. : } \mathbf{x} \in X$$

where $\rho > 0$ is a small augmentation coefficient and $\omega_1, \cdots, \omega_p$ are weights. Figure 2 shows how an **asf** projects reference points into the Pareto front. See [25] for more details about **asf**, the role of the weights, the augmentation coefficient, etc.

This is, **asf**s let you transform a **MOP** into a single-objective problem and obtain a single solution but adapted to the DM's preferences. But in most cases an iterative method is required to obtain the most preferred solutions as the DM learns about his/her preferences and the problem throughout the

interaction process, mainly changing the reference point or the weights in the **asf**. Then, an approximation of the Pareto front around the projected solution could be more interesting than simply the projected solutions, as a wider set of alternatives could be shown, all of them adapted to the DM's preferences, as shown in Figure 3.

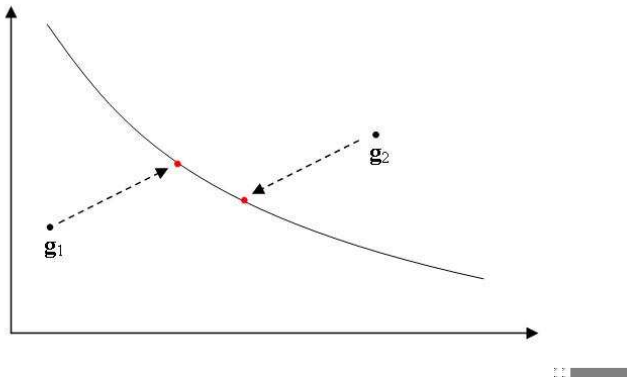

Fig. 2. Projection onto the Pareto front

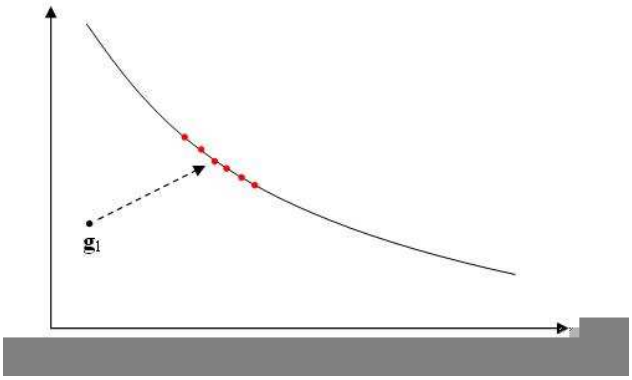

Fig. 3. Sample around the projection

As indicated before, this can be done by changing the reference point or the parameters in the **asf** and performing multiple runs. This requires the use of a single-objective optimizer instead of a multi-objective solver, and as a result a fixed number of solutions around the reference point are obtained. The main issue with this approach is how to manage the parameters in order to obtain a spread (but not too wide) approximation of the area of interest of the efficient front, this is, a representative sample of the area around the projection.

On the other hand, our proposal consists of modifying the Pareto dominance definition in order to directly obtain an approximation of the Pareto front around the projection using a multi-objective solver without setting or varying any parameter. Our proposed approach has the advantage of being very easy to implement and to couple into any MOMH. This aims to give the user the freedom of choosing the MOMH that considers as the most appropriate for the problem at hand, without having to worry about possible modifications to

the architecture of the search engine, as a requirement to incorporate his/her preferences.

4 g-dominance

Given a reference point $\mathbf{g} \in \mathbb{R}^p$ and a point $\mathbf{f} \in \mathbb{R}^p$, we define $Flag_{\mathbf{g}}(\mathbf{f})$ in the following way:

$$Flag_{\mathbf{g}}(\mathbf{f}) = \begin{cases} 1 & \text{if} \quad f_i \leq g_i \quad \forall i = 1, \cdots, p \\ 1 & \text{if} \quad g_i \leq f_i \quad \forall i = 1, \cdots, p \\ 0 & \text{otherwise} \end{cases}$$

This is, given a reference point $\mathbf{g}_1$, we divide the space in the following way (Fig. 4):

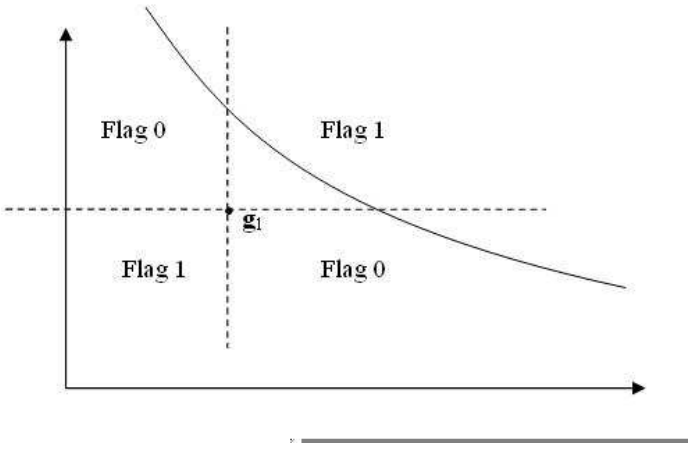

Fig. 4. Flags based on $\mathbf{g}_1$

And, based on these flags, we propose the following dominance relation (**g**-dominance):

Given two points **f**, **f'** $\in \mathbb{R}^p$, then, **f'** is **g**-dominated by **f** if:

1. $Flag_{\mathbf{g}}(\mathbf{f}) > Flag_{\mathbf{g}}(\mathbf{f'})$

or

2. Being $Flag_{\mathbf{g}}(\mathbf{f}) = Flag_{\mathbf{g}}(\mathbf{f'})$, we have:

$$f_i \leq f'_i \quad \forall i = 1, \cdots, p$$

with at least one j such that $f_j < f'_j$.

This will drive the search naturally to the desired area of the efficient front (it does not matter if the reference point is feasible or not), as shown in Figures 5 and 6:

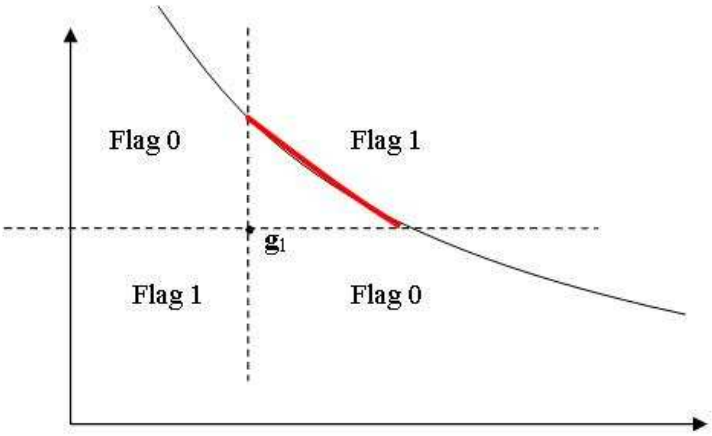

Fig. 5. Infeasible reference point

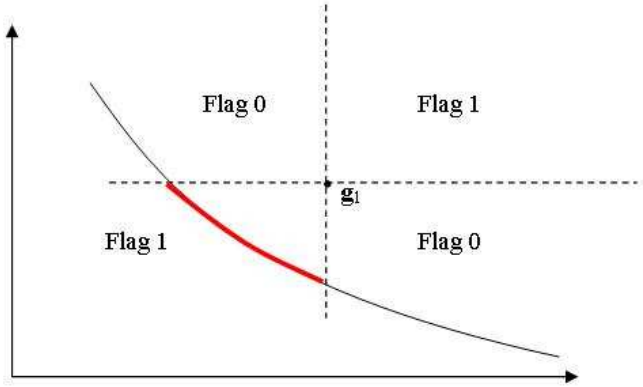

Fig. 6. Feasible reference point

Our proposed **g**-dominance can be easily implemented into any MOMH, by just changing the dominance-checking function or by changing the way in which the objective functions are evaluated. This last case is the most simple way to implement **g**-dominance in an existing code, as only requires the modification of the module evaluating the objective functions. For our problem (minimization) and given a reference point **g**, the **g**-dominance can be introduced evaluating the functions in the way shown in Algorithm 1, where M is a big number. This is based on a simple idea: penalize solutions with $Flag_{\mathbf{g}}(\mathbf{f}) = 0$ with a big amount M in order to make any solution with $Flag_{\mathbf{g}}(\mathbf{f}) = 0$ to be dominated by any solution with $Flag_{\mathbf{g}}(\mathbf{f}) = 1$. Computing the flags is very simple, too, as it is illustrated in Algorithm 2.

Algorithm 1 Function: evaluate $\mathbf{f}(x)$

Evaluate $f_i(x), \quad i = 1, \cdots, p$
Compute $Flag_{\mathbf{g}}(\mathbf{f})$
if $Flag_{\mathbf{g}}(\mathbf{f}) = 0$ **then**
    $f_i(x) = f_i(x) + M \qquad i = 1, \cdots, p$
end if

Algorithm 2 Function: Compute $Flag_{\mathbf{g}}(\mathbf{f})$

$Flag_{\mathbf{g}}(\mathbf{f}) = 1$
for $i = 1, \cdots, p$ **do**
    if $f_i(x) > g_i$ **then**
        $Flag_{\mathbf{g}}(\mathbf{f}) = 0$
    end if
end for
if $Flag_{\mathbf{g}}(\mathbf{f}) = 0$ **then**
    $Flag_{\mathbf{g}}(\mathbf{f}) = 1$
    for $i = 1, \cdots, p$ **do**
        if $f_i(x) < g_i$ **then**
            $Flag_{\mathbf{g}}(\mathbf{f}) = 0$
        end if
    end for
end if

This simple modification makes it possible to use **g**-dominance with any MOMH. In the next section we describe how to use the **g**-dominance integrated into a generic interactive scheme, in order to let the DM to iteratively achieve his/her most preferred solution.

*4.1 Using **g**-dominance in an interactive way*

Our proposed **g**-dominance can be used in an interactive scheme, where the DM will be guided iteratively to the most preferred solution. The way preferences are going to be included at each iteration is by changing the current reference point or by selecting a solution from the sample shown. Then, the DM will be shown a set of efficient solutions adapted to this new information provided. This is, the interaction will be carried out as shown in Algorithm 3, where, once a reference point $\mathbf{g}_t$ is provided at iteration t, the set of **g**-efficient solutions is called PF_{g_t}, and the sample from this set selected to be shown to the DM is called RS_t.

In other words, at each iteration, the DM is shown a set of solutions adapted to a reference point $\mathbf{g}_t$, and if he/she does not feel satisfied with any of these solutions, he/she can modify the reference point in order to refine the prefe-

Algorithm 3 Interaction

$t = 0$. Ask the DM to provide a reference point $\mathbf{g}_0$
while the DM is not satisfied **do**
  Compute the set PF_{g_t} of $\mathbf{g}_t$-efficient solutions.
  Select rs representative solutions from PF_{g_t}; $RS_t = \{s^1_{g_t}, \cdots, s^{rs}_{g_t}\}$.
  Show the set RS_t to the DM.
  if the DM is not satisfied with any of these solutions **then**
    if the DM wants to provide a new reference point $\mathbf{g}_{t+1}$ **then**
      Ask the DM to provide the new reference point $\mathbf{g}_{t+1}$.
    end if
    if the DM wants to select a solution in RS_t **then**
      Ask the DM to choose the most preferred solution in RS_t.
      Use this information to compute the new reference point $\mathbf{g}_{t+1}$.
    end if
    $t = t + 1$
  end if
end while

rences or he/she can select a solution $s^k_{g_t}$ in RS_t and a new reference point $\mathbf{g}_{t+1}$ will be computed using this information.

In this last case, the way to compute the new reference point $\mathbf{g}_{t+1}$ is by doing a convex combination of $s^k_{g_t}$ and $\mathbf{g}_{t+1}$:

$$\mathbf{g}_{t+1} = (1 - \theta) \cdot \mathbf{g}_t + \theta \cdot s^k_{g_t}$$

where θ is a parameter in $(0, 1)$ and represents the speed of convergence of the algorithm. The closer θ is to 1, the closest the new reference point is to $s^k_{g_t}$ and then the closest is the new set from $\mathbf{g}_{t+1}$-efficient solutions around $s^k_{g_t}$. This effect is shown in Figure 7.

The construction of RS_t is not trivial or simple. Some important questions arise, such as, for example, the number of solutions to include. Quite a lot of literature on Interactive Methods could be used to deal with these questions, and we consider it an interesting future research path.

The way we propose to select rs representative solutions from PF_{g_t}, is by using a clustering procedure. What we try to do here is to show the DM a number $(rs > p)$ of representative solutions from which to choose the most preferred ones. These rs reference solutions at iteration t will be the representative item of a cluster in PF_{g_t}. Given an iteration t and its corresponding set of solutions PF_{g_t}, the following procedure is used to choose the reference solutions:

Thus, this set contains a representative sample of PF_{g_t} including its p extreme points and $rs - p$ diverse compromise solutions. As mentioned above, this is

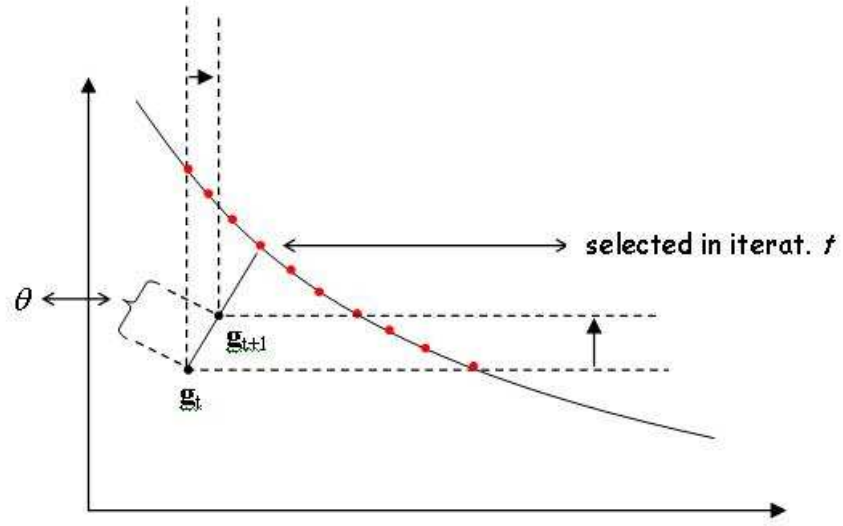

Fig. 7. New reference point

Algorithm 4 Building the set RS_t

for $i = 1, \cdots, p$ **do**
  Choose the best solution in PF_{g_t} for criteria i.
  Include it in RS_t.
end for
while $\#(RS_t) < rs$ **do**
  Choose the solution in $PF_{g_t} \backslash RS_t$ maximizing the distance from RS_t.
  Include it in RS_t.
end while

only one possible way to build this set, but many questions remain open at this point.

5 Computational Experiments

In order to validate our proposed approach, we coupled **g**-dominance to two different metaheuristics: the NSGA-II [13], which is a MOEA representative of the state-of-the-art in the area, and the DEMORS method [30], which is a hybrid of a differential evolution method with a Rough Sets tool. We used two test problems for our experiments: ZDT1 from the **ZDT** set [34] and deb32 from the **Deb** set [9]. Each problem is solved for three different reference points, feasible and infeasible.

Figures 8 and 9 show how both methods (i.e., the NSGA-II and DEMORS) are able to find a set of efficient points adapted to the information contained in the reference points. None of them required a deep modification in their

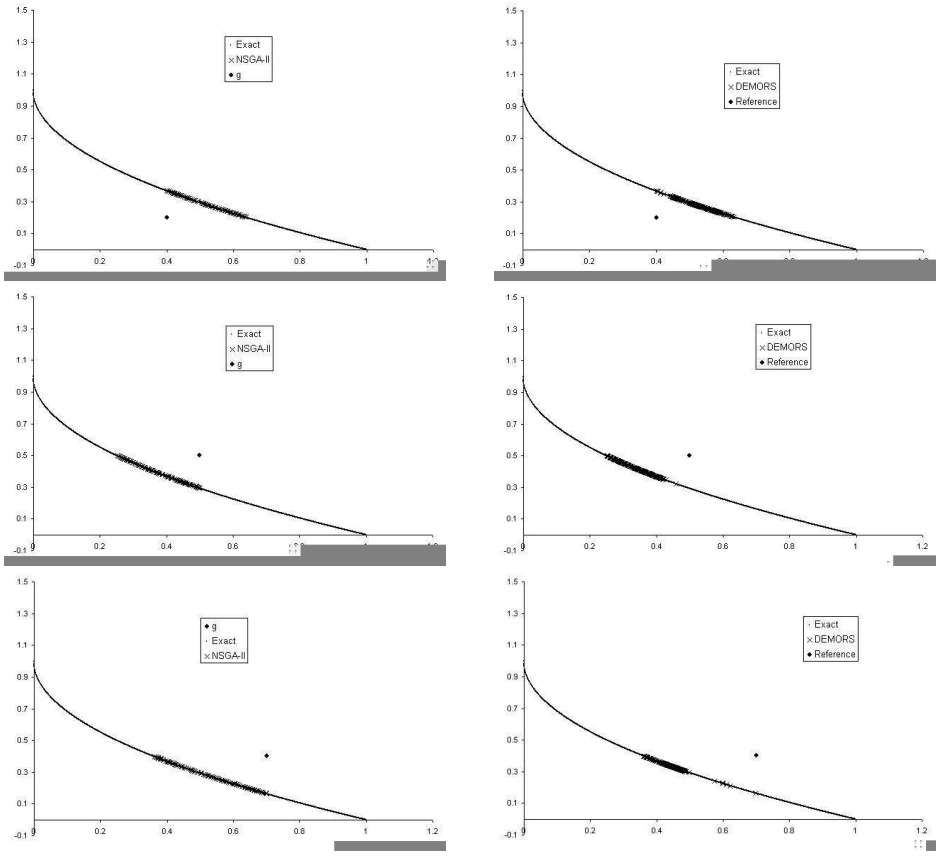

Fig. 8. Efficient solutions generated by the NSGA-II (left) and DEMORS(right) for the ZDT1 problem.

structure and they worked both for the feasible and the infeasible case.

6 Conclusions

In this paper, we propose a new concept of dominance, which we call **g**-dominance. This concept lets us approximate the efficient set around the area of the most preferred point without using any scalarizing function. This kind of dominance is independent of the MOMH used and can be easily coupled to any of them (either evolutionary or not) without any deep modification to the main structure of the method chosen.

We propose the use of **g**-dominance in an interactive scheme, where the DM is guided iteratively to the most preferred solution. The way preferences are to be included at each iteration is by changing the current reference point or by selecting a solution from the sample shown, and then the DM is shown a set of efficient solutions adapted to this new information provided. This kind of interaction is easy and intuitive for the DM and, together with the possibility of choosing any MOMH available, we believe that it may become an efficient

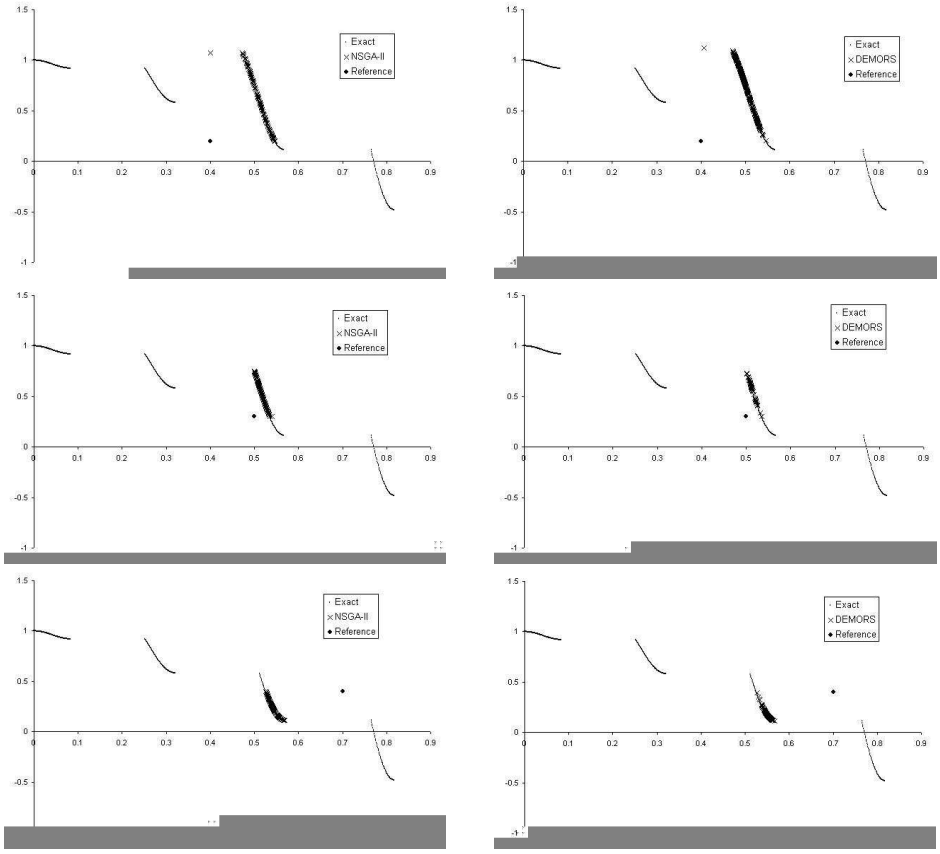

Fig. 9. Efficient solutions generated by the NSGA-II (left) and DEMORS(right) for the deb32 problem.

tool to deal with real-world problems.

On the other hand, some related aspects deserve a deeper analysis in the future. This is the case of the construction of the representative sample to be shown to the DM, or the performance of this approach when the number of objectives is increased.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments, which greatly helped them to improve the contents of this paper.

The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The fourth author acknowledges support from CONACyT through project number 45683-Y. This research has been partially funded too by the research projects of Andalusian Regional Government and Spanish Ministry of Education and Science.

References

[1] J. Branke and K. Deb. Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–477. Springer, Berlin Heidelberg, 2005. ISBN 3-540-22902-7.

[2] J. Branke, T. Kaußler, and H. Schmeck. Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software*, 32:499–507, 2001.

[3] C. A. Coello Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *2000 Congress on Evolutionary Computation*, volume 1, pages 30–37, Piscataway, New Jersey, July 2000. IEEE Service Center.

[4] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[5] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[6] D. Cvetković and I. C. Parmee. Genetic Algorithm–based Multi–objective Optimisation and Conceptual Engineering Design. In *Congress on Evolutionary Computation – CEC99*, volume 1, pages 29–36, Washington D.C., USA, 1999. IEEE.

[7] D. Cvetković and I. C. Parmee. Preferences and their Application in Evolutionary Multiobjective Optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1):42–57, February 2002.

[8] K. Deb. Multi–Objective Evolutionary Algorithms: Introducing Bias Among Pareto–Optimal Solutions. KanGAL report 99002, Indian Institute of Technology, Kanpur, India, 1999.

[9] K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.

[10] K. Deb. Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms. In *1999 Congress on Evolutionary Computation*, pages 77–84, Washington, D.C., July 1999. IEEE Service Center.

[11] K. Deb. Nonlinear goal programming using multi-objective genetic algorithms. *Journal of the Operational Research Society*, 52(3):291–302, 2001.

[12] K. Deb. Multi-objective Evolutionary Algorithms: Introducing Bias Among Pareto-optimal Solutions. In A. Ghosh and S. Tsutsui, editors, *Advances in Evolutionary Computing. Theory and Applications*, pages 263–292. Springer, Berlin, 2003.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[14] K. Deb, J. Sundar, U. B. R. N., and S. Chaudhuri. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286, 2006.

[15] M. Ehrgott and X. Gandibleux. A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization. *OR Spektrum*, 22:425–460, 2000.

[16] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[17] G. W. Greenwood, X. S. Hu, and J. G. D'Ambrosio. Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 437–455, San Mateo, California, 1997. Morgan Kaufmann.

[18] M. Hapke, A. Jaszkiewicz, and R. Slowinski. Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research*, 107(2):315–324, 1998.

[19] A. Jaszkiewicz and R. Slowinski. The Light Beam Search approach—an overview of methodology and applications. *European Journal of Operational Research*, 113(2):300–314, 1999.

[20] D. Jones, S. Mirrazavi, and M. Tamiz. Multi-objective metaheuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, February 2002.

[21] P. Korhonen and J. Laakso. A Visual Interactive Method for Solving the Multiple Criteria Problem. *European Journal of Operational Research*, 24(2):277–287, 1986.

[22] P. Korhonen and J. Wallenius. A Pareto Race. *Naval Research Logistics*, 35(6):615–623, 1988.

[23] S. Massebeuf, C. Fonteix, L. N. Kiss, I. Marc, F. Pla, and K. Zaras. Multicriteria Optimization and Decision Engineering of an Extrusion Process Aided by a Diploid Genetic Algorithm. In *1999 Congress on Evolutionary Computation*, pages 14–21, Washington, D.C., July 1999. IEEE Service Center.

[24] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3):909–922, 2006.

[25] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[26] H. Nakayama and Y. Sawaragi. Satisficing trade-off method for multiobjective programming. In M. Grauer and A. Wierzbicki, editors, *Interactive Decision Analysis. Proceedings of an International Workshop on Interactive Decision Analysis and Interpretative Computer Intelligence*, pages 113–122. Springer-Verlag. Lecture Notes in Economics and Mathematical Systems Vol. 229, 1984.

[27] S. Phelps and M. Koksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12):1726–1738, December 2003.

[28] M. João Alves and João Clímaco. An Interactive Method for 0-1 Multiobjective Problems Using Simulated Annealing and Tabu Search. *Journal of Heuristics*, 6(3):385–403, August 2000.

[29] B. Rekiek, P. D. Lit, F. Pellichero, T. L'Eglise, E. Falkenauer, and A. Delchambre. Dealing With User's Preferences in Hybrid Assembly Lines Design. In *Proceedings of the MCPL'2000 Conference*, 2000.

[30] L. V. Santana-Quintero, N. Ramírez-Santiago, C. A. Coello Coello, J. Molina Luque, and A. G. Hernández-Díaz. A New Proposal for Multiobjective Optimization Using Particle Swarm Optimization and Rough Sets Theory. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 483–492. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.

[31] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley, New York, 1986.

[32] E. Ulungu, J. Teghem, and C. Ost. Efficiency of interactive multi-objective simulated annealing through a case study. *Journal of the Operational Research Society*, 49:1044–1050, 1998.

[33] A. P. Wierzbicki. The Use of Reference Objectives in Multiobjective Optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, pages 469–486. Springer-Verlag, New York, 1980.

[34] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.