Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods

Cristóbal Barba-González[1], Vesa Ojalehto[2], José García-Nieto[1(✉)], Antonio J. Nebro[1,2], Kaisa Miettinen[2], and José F. Aldana-Montes[1]

[1] Dep. Lenguajes y Ciencias de la Computación, Ada Byron Research Building, University of Málaga, 29071 Málaga, Spain
{cbarba,jnieto,antonio,jfam}@lcc.uma.es

[2] Faculty of Information Technology, University of Jyvaskyla, P.O. Box 35, 40014 Agora, Finland
{vesa.ojalehto,kaisa.miettinen}@jyu.fi

Abstract. Over the years, many interactive multiobjective optimization methods based on a reference point have been proposed. With a reference point, the decision maker indicates desirable objective function values to iteratively direct the solution process. However, when analyzing the performance of these methods, a critical issue is how to systematically involve decision makers. A recent approach to this problem is to replace a decision maker with an artificial one to be able to systematically evaluate and compare reference point based interactive methods in controlled experiments. In this study, a new artificial decision maker is proposed, which reuses the dynamics of particle swarm optimization for guiding the generation of consecutive reference points, hence, replacing the decision maker in preference articulation. We use the artificial decision maker to compare interactive methods. We demonstrate the artificial decision maker using the DTLZ benchmark problems with 3, 5 and 7 objectives to compare R-NSGA-II and WASF-GA as interactive methods. The experimental results show that the proposed artificial decision maker is useful and efficient. It offers an intuitive and flexible mechanism to capture the current context when testing interactive methods for decision making.

Keywords: Multiobjective optimization · Preference articulation
Multiple criteria decision making · Particle swarm optimization

1 Introduction

Interactive multiobjective optimization methods based on a reference point are very popular techniques [1–3] not only in current research, but also in industry, as they allow decision makers (DMs) to specify information about their preferences in an intuitive manner to direct the operation of the optimization algorithms. As a consequence, the DM is able to learn progressively (at each iteration) about the

A. Auger et al. (Eds.): PPSN 2018, LNCS 11101, pp. 274–285, 2018.
https://doi.org/10.1007/978-3-319-99253-2_22

set of (approximated) solutions in the Pareto front of a complex problem, hence reducing one's cognitive load [2]. A second advantage of applying interactive multiobjective optimization methods is that they only need to generate those solutions interesting for the DM, i.e., that are in the region of interest.

Nevertheless, a critical issue arises when testing and comparing interactive methods [1,4], since they require the DMs to be involved in the solution process. Therefore, this involvement makes experiments much more costly than testing by computational means. In addition, other human factors take part, such as inconsistency and variability among decisions, learning curve when facing problems, and different times in solution processes.

In order to cope with this deficiency, a useful approach is to use artificial DMs (ADMs) as mechanisms to generate preference information when comparing interactive methods. Because interactive methods utilize different types of preference information [3,5], appropriate ADMs are demanded for each type. Indeed, in comparison with the amount and diversity of existing interactive methods, the number of ADMs is limited [1]. Interactive methods can be divided into non ad hoc and ad hoc methods depending on whether the DM can be replaced by a value function or not, respectively (see, e.g., [1,6]). Reference point based methods belong to the latter group. However, many popular interactive methods are based on reference points [2,3], where the DM represents the region of interest as a vector of desirable objective values.

Recently, in [7] a new ADM has been developed for testing reference point based interactive methods. It is able to adjust reference points based on information about solutions derived so far. The adjustment involves randomness and the amount of noise decreases during the interactive solution process. The overall procedure is based on a pre-defined neighborhood of a most preferred solution.

Following this line of research, a novel ADM is proposed here that reuses the dynamics of particle swarm optimization (PSO) to guide the generation of reference points, hence, replacing the DM in preference articulation. The idea is to derive reference points by particle's movements in the swarm, which evolves in the objective space. The main contributions of the proposed ADM in this paper are as follows:

- It offers an intuitive, bio-inspired and flexible mechanism to capture the current context in interactive solution processes when tackling multiobjective optimization problems. At each iteration of the process, nondominated solutions derived so far can be used in generating the new reference point.
- It avoids dependence on the pre-defined target levels for objectives.
- It allows different parameter settings to enhance diversification/intensification in the generation of new reference points.

The new ADM is tailored for comparing interactive evolutionary reference point based methods. We demonstrate it on the DTLZ benchmark problems with 3, 5 and 7 objectives and two reference point evolutionary methods R-NSGA-II [8] and WASF-GA [9]. Thus, we use them as examples of interactive EMOs (iEMOs). The experimental results show that the proposed ADM is useful and efficient when compared to the previous one.

The rest of this paper is organized as follows. Section 2 contains background concepts and related work. The proposed ADM is described in Sect. 3. Section 4 summarizes experimental results, analysis and discussions. Finally, conclusions and lines of future work are outlined in Sect. 5.

2 Background

Evolutionary multiobjective optimization methods have been shown to perform successfully when finding a set of trade-off solution approximations representing a Pareto front to complex multiobjective optimization problems. Nevertheless, a common requirement in real-world problems arises in solution process where not only Pareto front approximations are demanded, but it is desirable to find preferred solutions or regions that reflect human DM's desires or tendencies.

Interactive methods are able to focus on an area of interest in the objective space, in order to find preferred solutions [1]. Examples of ways how a DM can provide preference information are comparisons of small sets of solutions, classification or indicating desired trade-offs [1,3,6]. Furthermore, as mentioned in the introduction, an intuitive type of preference articulation in interactive methods is based on reference points [2,3], which consist of desirable objective function values.

The difficulty arises when trying to evaluate and compare interactive methods based on reference points, since a human DM is required to take part in the solution process to specify reference points. On the other hand, as stated in [4], there exists a strong necessity of creating automatic DMs to facilitate the comparison of different methods.

We consider multiobjective optimization problems of the form

$$\begin{aligned} &\text{minimize} \quad && \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \\ &\text{subject to} \quad && \mathbf{x} = (x_1, \ldots, x_n)^T \in S, \end{aligned} \tag{1}$$

where we minimize[1] k ($k \geq 2$) objective functions $f_i : S \rightarrow \mathbb{R}$ on the set $S \subset \mathbb{R}$ of feasible solutions (decision vectors). The elements in the objective space $\mathbb{R}^k$ are the objective (function) values $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$, usually called *objective vectors*. We denote the set of feasible objective vectors by $Z = \mathbf{f}(S)$. The so-called *Pareto optimal set* of solutions to the problem is defined as:

$$E = \left\{ \mathbf{x} \in S : \nexists\, \mathbf{x}^{'} \in S \mid f_i(\mathbf{x}^{'}) \leq f_i(\mathbf{x}),\ i = 1, \cdots, k \ \text{ and } \ \mathbf{f}(\mathbf{x}^{'}) \neq \mathbf{f}(\mathbf{x}) \right\} \tag{2}$$

and the corresponding objective vectors form a Pareto front.

Artificial Decision Maker: In what follows, we refer to the ADM proposed in [7] as the original ADM. It consists of three main components: steady part, current context and preference information. We need the concepts of *ideal* ($\mathbf{z}^*$) and *nadir* ($\mathbf{z}^{\mathbf{nad}}$) objective vectors of the problem to find reference points.

[1] Without loss of generality, we use minimization in definitions.

The former is defined as $\mathbf{z}^* = (z_1^*, \ldots, z_k^*)^T$, where $z_i^* = min_{x \in S} f_i(\mathbf{x})$ for $i = 1, \ldots, k$, whereas the later is defined as $\mathbf{z^{nad}} = (z_1^{nad}, \ldots, z_k^{nad})^T$, where $z_i^{nad} = max_{x \in E} f_i(x)$ for $i = 1, \ldots, k$. If these vectors are not known a priori, the ideal objective vectors can be calculated and the nadir estimated [3]. When applying iEMOS, they can e.g. be estimated from the current population. The three main components of the ADM are:

- *Steady part*: This part includes experience and knowledge available at the beginning of the solution process and remains unchanged in the solution process. As an example, the steady part can consist of a region of interest or of target levels specific to objective functions that are desired to be achieved.
- *Current context*: This part includes all the knowledge about the problem which is gained during the solution process by the ADM, for instance, shape of the Pareto front, trade-offs between the objectives, obtainable objective function values (e.g., $\mathbf{z}^*$ and $\mathbf{z^{nad}}$), etc.
- *Preference information*: With this part, the ADM expresses its knowledge during the solution process in order to guide the method towards solutions that are more preferred by the ADM. Preference information is method-specific and in this research we consider reference points $\mathbf{q} = (q_1, \ldots, q_k)^T$.

3 Artificial Decision Maker Driven by PSO

As mentioned before, we propose an ADM that enables testing interactive methods, where preference information is given in the form of a reference point. The proposed ADM utilizes PSO in modifying the current context of the original ADM and we call it ADM-PSO.

Given an iteration counter t, a *reference point* is denoted by $\mathbf{q}_t = (q_{t,1}, \ldots, q_{t,k})^T$. It is said to be *achievable* for problem (1), if $\mathbf{q}_t \in Z + \mathbb{R}^k_+$ (where $\mathbb{R}^k_+ = \{\mathbf{y} \in \mathbb{R}^k \mid y_i \geq 0 \text{ for } i = 1, \ldots, k\}$), that is, if either $\mathbf{q}_t \in Z$ or if $\mathbf{q}_t$ is dominated by a Pareto optimal objective vector in Z. Otherwise, the reference point is said to be *unachievable*, that is, not all of its components can be achieved simultaneously.

By using a reference point $\mathbf{q}_t$ in the iteration t, an ADM is able to feed an interactive multiobjective optimization method with preferences. Then the method can direct the solution process accordingly. If the method is evolutionary, it can generate approximations of Pareto optimal solutions oriented to this specific region of interest. This new set of nondominated solutions can be in turn used to generate a new reference point $\mathbf{q}_{t+1}$ for the next iteration of the method. This process can be repeated until a stopping criterion is valid. In our case, we use a pre-defined point **asp** to be called ADM-aspiration point and stop once we get a reference point close enough to it. Intuitively, an additional (single objective) optimization problem arises in this process, since the new reference point is to be generated, with a minimum distance to **asp**. In our case, the current Pareto front approximation is used as a population to train the implicit learning model of the optimization method, i.e., the ADM. In this way, the new ADM is able to operate on the objective space by taking advantage of all the information provided by the interactive method.

Keeping this idea in mind, the proposed approach focuses on the use of a canonical PSO to carry out the generation of new reference points, hence acting as an ADM which is able to interact with the underlying interactive multiobjective optimization method. As mentioned earlier, in this study we consider iEMO methods. The aim is to reuse the biological inspiration modeling a particle's dynamics in PSO, to replace DMs when managing their preferences.

A conceptual sketch of this approach is illustrated in Fig. 1, where the new reference point $\mathbf{q}_{t+1}$ is generated in one movement step of the PSO. It takes into account the previous reference point $\mathbf{q}_t$ as well as the objective vectors of the nondominated solutions in the current Pareto front approximation, provided by the underlying iEMO.

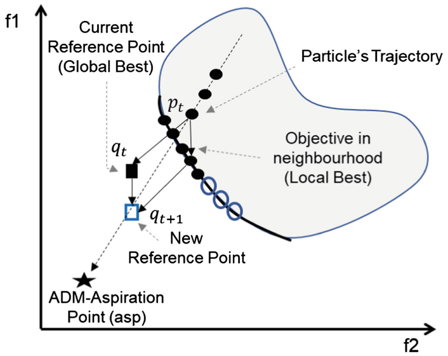

Fig. 1. Conceptual sketch of an ADM-PSO operation. The new reference point $\mathbf{q}_{t+1}$ is generated by means of PSO particle's movement operators.

Among the many existing PSO variants, for simplicity, ADM-PSO is based on the standard version 2007 [10]. It provides the canonical equations to model the particle's movements, which have been adapted to cope with the reference point generation as follows: Each particle's position vector $\mathbf{p}$ (codifying an objective vector) is updated at each iteration t as

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{v}_{t+1}, \tag{3}$$

where $\mathbf{p}_{t+1}$ is a new candidate reference point ($\mathbf{p}_{t+1} = \mathbf{q}_{t+1}$) and $\mathbf{v}_{t+1}$ is the velocity vector of the particle given by

$$\mathbf{v}_{t+1} = \omega \cdot \mathbf{v}_t + U^t[0, \varphi_1] \cdot (\mathbf{l}_t - \mathbf{p}_t) + U^t[0, \varphi_2] \cdot (\mathbf{b}_t - \mathbf{p}_t). \tag{4}$$

In (4), $\mathbf{l}_t$ is the local best position the particle $\mathbf{p}_t$ has ever stored and $\mathbf{b}_t$ is the position found by the member of its neighborhood that has had the best performance so far. In ADM-PSO, $\mathbf{b}_t = \mathbf{q}_t$, i.e., it is set as the reference point.

Acceleration coefficients φ_1 and φ_2 control the relative effect of the personal and social best particles and U^t is a diagonal matrix with elements distributed in the interval $[0, \varphi_i]$, uniformly at random. Finally, $\omega \in (0, 1)$ is called the inertia weight and influences the trade-off between exploitation and exploration. These parameters can be used to induce additional preference information into the ADM. In particular, ADM-PSO is able to set the *current context* (defined in Sect. 2) by using not only the nearest point to **asp** (as done by the original ADM), but all the points (objective vectors of nondominated solutions) in the Pareto front approximation provided by the iEMO. Consequently, this allows the ADM-PSO to explore thoroughly the Pareto front in the objective space.

In order to asses the adequacy of the new generated reference points, the following single objective fitness function is used by ADM-PSO:

$$d(\mathbf{x_q}) = \sqrt{\sum_{i=1}^{k}(f_i(\mathbf{x_q}) - asp_i)}. \tag{5}$$

In short, the function $d(\mathbf{x_q})$ calculates the Euclidean distance between the nearest point (solution $\mathbf{x_q}$) of the Pareto front approximation obtained with the reference point $\mathbf{q}$, and the point **asp**, where k is the number of objectives. As commented before, ADM-PSO aims at minimizing this distance.

Algorithm of ADM-PSO

For the sake of a better understanding, the pseudo-code of ADM-PSO is shown in Algorithm 1. The first phase corresponds to initialization of parameters, populations and initial Pareto set approximations (from line 1 to 11). In this phase, an initial reference point is also generated (line 12) as done in the original ADM (see Sect. 3 in [7]). After this, the iterative solution process (line 13) starts with multiple rounds of the interactive multiobjective optimization method (line 14) and the corresponding generation of new reference points, by means of PSO (line 16). Each ADM round (lines 13-18) entails a maximum number of iterations (I_{max}) in which the iEMO algorithm in question is run until reaching a maximum number of generations G_{max} (line 14). The PSO is then invoked to obtain a new reference point, which uses the last obtained Pareto set approximation from the previous step. Before that, an intermediate step (line 15) is computed to "accommodate" objective vectors in the Pareto front approximation (or nondominated points) to the swarm (S_{t+1}). At the end, the approximation of the region of interest found is returned (line 19) and the whole algorithm ends.

ADM-PSO has been developed in the jMetal library of EMOs and following its architectural style [11] with the aim of taking advantage of all the functionalities provided in this framework: solution types, operators, algorithms, problems, etc. It is worth noting that the core algorithm has been designed to provide a general (software) template, so that iEMOs to be tested can be easily configured. As mentioned, the current configuration contains iEMOs R-NSGA-II and WASF-GA. In this way, a framework for the evaluation and comparison of iEMOs is available[2].

[2] https://github.com/KhaosResearch/admpso.

Algorithm 1. Pseudo-code of ADM-PSO

I_{max} // Maximum number of ADM iterations
G_{max} // Maximum number of iEMO generations
c, m // Genetic operators
$t \leftarrow 0$ // ADM iteration counter
A // Multiobjective optimization problem
S // Maximum swarm size of ADM-PSO
φ_1, φ_2, ω // PSO specific parameters
M // iEMO algorithm(s) tested
$P_t \leftarrow initializePopulation(N)$ // where N is Population size
$evaluate(P_t, A)$
$E_t \leftarrow initializeParetoSet(P_t)$
$\mathbf{q}_t \leftarrow initializeReferencePoint(\mathbf{asp}, \mathbf{z}^*, \mathbf{z}^{nad}, w_r, p_r)$ // As in original ADM
while $(t < I_{max})$ AND $(\mathbf{asp} \neq \mathbf{q}_t)$ **do**
    $(P_{t+1}, E_{t+1}) \leftarrow compute_{iEMO}(M, \mathbf{q}_t, c, m, P_t, A, G_{max})$ // Evolves iEMO
    $S_{t+1} \leftarrow setNewSwarm(S, E_{t+1})$ // Generate new swarm from E_{t+1}
    $\mathbf{q}_{t+1} \leftarrow compute_{PSO}(\mathbf{asp}, \mathbf{q}_t, S_{t+1}, \varphi_1, \varphi_2, \omega)$ // Generate new reference point
    $t \leftarrow t + 1$
end while
return E_{t+1} // Notify Pareto front approximation

4 Experimental Results

In order to demonstrate the validity of the proposed approach, a series of experiments has been conducted to test two iEMOs called WASF-GA [9] and R-NSGA-II [8]. In the experiments, ADM-PSO generates reference points for the methods, hence enabling automatic tests and comparisons. For these experiments, a common framework has been used that comprises of a family of seven DTLZ benchmark problems [12] with 3, 5 and 7 objectives, summing up to 21 different problems. For each combination of algorithms and problems, 31 independent runs were performed.

In these experiments, a set of fixed ADM-aspiration points (asp) was configured for each problem. They are all achievable and calculated by taking into account the estimated ideal and nadir objective vectors for each problem as $asp_i = 2/3 \times z_i^{nad} + z_i^*$. It is worth noting that for these problems the ideal objective vectors are always at the origin $(0, \ldots, 0)$, whereas nadir objective vectors were obtained from the worst solutions (ranges) found so far in preliminary experiments, where algorithmic parameters were tuned as described below. In this regard, Table 1 shows the nadir objective vectors used with the corresponding asp for each problem, as well as the number of objective functions (k).

In order to enable fair comparisons, WASF-GA and R-NSGA-II were set using a common parameter setting that consists of a population size $N = 100$, external archive size $E = 100$, a maximum number of (iEMO) generations $G_{max} = 20,000$, a crossover SBX with a probability $c = 0.9$ and a distributional index 20, a polynomial mutation with a probability $m = 0.1$, a mutation distributional index 20, and a binary tournament selection. In the case of R-NSGA-II, the epsilon parameter was set to 0.0045.

Table 1. Achievable ADM-aspiration points (*asp*) and nadir objective vectors used.

Problem	k	**asp**	$\mathbf{z}^{\mathbf{nad}}$
DTLZ1	3	(6.7, 26.7, 133.4)	(10.0, 40.0, 200.0)
	5	(7.0, 26.7, 133.7, 33.6, 100.4)	(10.0, 40.0, 200.5, 50.5, 150.5)
	7	(7.0, 26.7, 133.7, 33.6, 100.4, 31.0, 67.7)	(10.0, 40.0, 200.5, 50.5, 150.5, 46.5, 101.5)
DTLZ2	3	(2.6, 1.4, 1.4)	(4.0, 2.0, 2.0)
	5	(2.6, 1.4, 1.4, 1.4, 1.4)	(4.0, 2.0, 2.0, 2.0, 2.0)
	7	(2.6, 1.4, 1.4, 1.4, 1.4, 1.4, 1.4)	(4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0)
DTLZ3	3	(17.0, 122.0, 38.7)	(25.5, 183.0, 58.0)
	5	(1.0, 19.0, 1.0, 667.0, 668)	(1.5, 28.5, 1.5, 999.0, 999.5)
	7	(17.0, 122.0, 40.0, 40.0, 667.0, 668.0, 667.0)	(25.5, 183.0, 60.0, 60.0, 999.0, 999.5, 999.0)
DTLZ4	3	(1.4, 1.4, 1.4)	(2.0, 2.0, 2.0)
	5	(1.4, 1.4, 1.4, 1.4, 1.4)	(2.0, 2.0, 2.0, 2.0, 2.0)
	7	(1.4, 1.4, 1.4, 1.4, 1.4, 1.4, 1.4)	(2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0)
DTLZ5	3	(1.4, 1.4, 1.4)	(2.0, 2.0, 2.0)
	5	(1.4, 1.4, 3.0, 3.0, 1.7)	(2.0, 2.0, 3.0, 3.0, 2.5)
	7	(1.4, 1.4, 1.4, 1.4, 1.0)	(2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 1.5)
DTLZ6	3	(3.0, 2.7, 4.4)	(4.5, 4.0, 6.5)
	5	(3.0, 2.7, 4.4, 4.4, 5.4)	(4.5, 4.0, 6.5, 6.5, 8.0)
	7	(3.0, 2.7, 4.4, 4.4, 4.4, 3.7, 3.4)	(4.5, 4.0, 6.5, 6.5, 6.5, 5.5, 5.0)
DTLZ7	3	(1.4, 1.4, 13.4)	(2.0, 2.0, 20.0)
	5	(1.4, 1.4, 1.4, 1.4, 21.7)	(2.0, 2.0, 2.0, 2.0, 32.5)
	7	(1.4, 1.4, 1.4, 1.4, 1.4, 1.4, 42.7)	(2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 55.0)

For ADM-PSO, parameters were set by following the previous work and standard settings of PSO 2007 [10]. It comprised of a maximum number of iterations $I_{max} = 11$, an objective consideration probability $p = 0.5$, a weight $w = 1/k$ and a tolerance $\theta = 10^{-3}$ (see [7] for a further explanation of ADM parameters). For PSO, we set $\varphi_1 = \varphi_2 = 1/(2 + \log(2))$ and inertia $\omega = 1/(2 \cdot \log(2))$. Since the swarm is fed with those non-dominated points of the external archive of the iEMO, the maximum swarm size was set accordingly, i.e., $S = E = 100$ each time the ADM-PSO is started. It conducted only 3 generations to assure that particles are able to move accordingly with new data.

In addition, with the aim of keeping track of the original ADM [7], it was also applied by following the same procedure. The results of ADM and ADM-PSO are arranged in Tables 2 and 3, respectively. In these tables, WASF-GA and R-NSGA-II are compared using the DTLZ problems, where for brevity the number of objectives is limited to 3, 5 and 7. The mean, standard deviation (STD) and minimum (MIN) distances of the nearest point (in the resulting Pareto front approximation) to the ADM-aspiration point *asp* are reported, together with the number of iterations (ITER) the ADM used on the average.

The first observation can be made from Tables 2 and 3 with regards to ADM and ADM-PSO. They performed in a similar way when guiding the underlying iEMOs (WASF-GA and R-NSGA-II) to find solutions close to the ADM-aspiration point. In this sense, no statistical differences could be found when comparing the mean distance distributions of all the combinations of ADMs with iEMOs. To be more specific, according to Friedman's test [13] with χ^2 and 3 degrees of freedom, a value of 6.07 was obtained ($<$7.81 from a χ^2 distribution table $\alpha = 0.05$), so the null hypothesis could not be rejected.

Table 2. WASF-GA versus R-NSGA-II with the original ADM.

Problem	k	WASF-GA				R-NSGA-II			
		MEAN	STD	MIN	ITER	MEAN	STD	MIN	ITER
DTLZ1	3	3.54E-02	3.33E-02	1.08E-02	7.70E+00	4.53E-01	1.57E-01	2.46E-01	8.40E+00
DTLZ1	5	1.60E+00	4.82E-01	4.83E-01	8.10E+00	2.15E+00	2.48E-01	1.56E+00	8.00E+00
DTLZ1	7	3.15E-01	1.48E-01	8.95E-02	9.00E+00	7.33E-02	2.82E-02	4.16E-02	7.50E+00
DTLZ2	3	1.75E-01	3.84E-01	2.42E-02	9.40E+00	7.11E-02	3.33E-02	1.82E-02	8.40E+00
DTLZ2	5	4.24E-01	3.14E-01	2.23E-01	7.30E+00	3.90E-01	6.22E-02	3.01E-01	7.90E+00
DTLZ2	7	1.68E-01	4.22E-02	1.10E-01	6.60E+00	1.43E-01	9.09E-02	4.07E-02	8.80E+00
DTLZ3	3	4.78E+00	1.23E+00	3.41E+00	7.00E+00	1.29E+01	2.33E+00	1.02E+01	7.00E+00
DTLZ3	5	3.74E+01	4.41E+00	2.91E+01	7.70E+00	2.77E+01	2.17E+01	1.06E+00	9.50E+00
DTLZ3	7	2.23E+02	3.33E+01	1.73E+02	6.60E+00	3.25E+02	2.78E+01	2.81E+02	8.10E+00
DTLZ4	3	6.01E-01	5.16E-04	6.01E-01	9.20E+00	6.06E-01	3.92E-03	6.02E-01	8.00E+00
DTLZ4	5	6.97E-01	1.17E-01	5.49E-01	7.30E+00	3.70E-01	6.40E-02	3.18E-01	6.90E+00
DTLZ4	7	7.03E-01	2.64E-02	6.51E-01	8.10E+00	6.25E-01	2.81E-02	5.80E-01	8.20E+00
DTLZ5	3	1.70E-01	3.16E-04	1.70E-01	9.10E+00	1.62E-01	5.86E-03	1.51E-01	7.90E+00
DTLZ5	5	9.51E-03	5.27E-03	4.92E-03	9.30E+00	1.29E-01	2.69E-02	9.67E-02	7.10E+00
DTLZ5	7	1.51E-01	2.54E-02	1.15E-01	7.70E+00	1.13E-01	3.74E-02	4.85E-02	7.00E+00
DTLZ6	3	4.52E-01	4.72E-01	1.60E-01	7.20E+00	1.48E+00	1.29E-01	1.25E+00	8.10E+00
DTLZ6	5	1.46E+00	1.25E+00	1.52E-01	7.50E+00	2.83E+00	1.37E+00	9.00E-01	5.20E+00
DTLZ6	7	4.87E+00	1.52E-01	4.63E+00	7.70E+00	4.60E+00	1.72E-01	4.20E+00	8.60E+00
DTLZ7	3	2.16E-02	1.96E-02	4.12E-03	9.90E+00	4.85E-01	1.10E-01	2.92E-01	7.60E+00
DTLZ7	5	2.90E+00	6.02E-01	1.88E+00	7.10E+00	2.80E+00	7.16E-01	1.75E+00	8.80E+00
DTLZ7	7	2.05E+01	1.52E+00	1.78E+01	7.10E+00	1.24E+01	9.65E-01	1.06E+01	4.70E+00

Nevertheless, ADM-PSO was able to obtain solutions in a lower number of iterations, which means an advantage in the computational effort. This can be observed in columns ITER of Tables 2 and 3, where ADM-PSO with WASF-GA used a lower number of iterations than the original ADM with WASF-GA in 15 out of 21 problems. Furthermore, ADM-PSO with R-NSGA-II required fewer iterations than the original ADM with R-NSGA-II for all the problems except for DTLZ7 with 7 objectives. Overall, the number of iterations can be used as an indicator for the solution process even though a smaller number does not directly mean a good performance. The ADM may be e.g. tailored to focus first on learning where very different reference points are used for scanning the Pareto front.

In this sense, it is worth noting that the aim of iEMOs is not to obtain a complete coverage of the Pareto front, but to focus on a specific region of interest relevant for a DM. WASF-GA and R-NSGA-II usually generate a set of solutions in that region, so that ADM-PSO uses them when forming the swarm. In this way, it is able to take advantage of information in the current context while experimenting a fast convergence (typical in PSO) to the ADM-aspiration point, i.e., to generate better reference points.

A special case was registered for problem DTLZ3 in Tables 2 and 3 since the performances of the iEMOs involved usually deteriorated as the ADM did not achieve the ADM-aspiration points consistently. Probably, the heterogeneity in the ranges observed when calculating the nadir objective vectors for this problem made the ADMs to generate the points *asp* close to unachievable regions, hence leading the iEMO to require extra effort to reach it.

In general, the performance of the iEMOs got worse as the number of objective functions increased. This is not surprising, since the complexity of DTLZ

Table 3. WASF-GA versus R-NSGA-II with ADM-PSO.

Pro blem	k	WASF-GA				R-NSGA-II			
		MEAN	STD	MIN	ITER	MEAN	STD	MIN	ITER
DTLZ1	3	1.29E-02	5.41E-03	4.69E-03	6.80E+00	2.01E-01	1.34E-01	6.06E-02	5.80E+00
DTLZ1	5	1.73E+00	3.39E-01	1.11E+00	7.90E+00	2.18E+00	2.01E-01	1.95E+00	5.30E+00
DTLZ1	7	1.89E-01	7.71E-02	6.18E-02	7.60E+00	6.70E-02	1.37E-02	4.64E-02	6.10E+00
DTLZ2	3	2.50E-02	5.67E-04	2.44E-02	8.10E+00	5.51E-02	2.62E-02	2.98E-02	6.60E+00
DTLZ2	5	4.65E-01	3.75E-01	1.26E-01	6.00E+00	4.44E-01	1.92E-01	3.62E-01	6.40E+00
DTLZ2	7	1.51E-01	4.08E-02	1.06E-01	7.00E+00	9.11E-02	9.77E-02	2.88E-02	7.00E+00
DTLZ3	3	4.00E+00	1.28E+00	1.70E+00	7.50E+00	1.02E+01	2.35E+00	6.78E+00	6.10E+00
DTLZ3	5	2.59E+01	1.15E+01	1.35E+01	5.90E+00	4.60E+00	2.59E+00	1.88E+00	7.20E+00
DTLZ3	7	1.95E+02	5.60E+01	1.22E+02	7.50E+00	3.30E+02	1.78E+01	2.98E+02	7.80E+00
DTLZ4	3	6.01E-01	0.00E+00	6.01E-01	7.20E+00	6.03E-01	3.07E-03	6.01E-01	4.60E+00
DTLZ4	5	4.39E-01	4.80E-02	3.24E-01	8.30E+00	3.17E-01	9.62E-03	3.03E-01	6.80E+00
DTLZ4	7	6.99E-01	3.37E-02	6.27E-01	7.30E+00	5.87E-01	6.96E-03	5.77E-01	7.30E+00
DTLZ5	3	1.70E-01	2.93E-17	1.70E-01	6.10E+00	1.64E-01	3.84E-03	1.56E-01	7.10E+00
DTLZ5	5	8.96E-03	3.92E-03	2.54E-03	7.00E+00	1.14E-01	2.75E-02	8.81E-02	6.00E+00
DTLZ5	7	1.25E-01	2.48E-02	9.47E-02	5.10E+00	7.54E-02	2.44E-02	4.22E-02	6.20E+00
DTLZ6	3	2.57E-01	6.94E-02	1.54E-01	6.80E+00	1.47E+00	1.16E-01	1.24E+00	7.60E+00
DTLZ6	5	1.07E+00	1.37E+00	1.54E-01	8.30E+00	2.51E+00	1.02E+00	1.29E+00	5.10E+00
DTLZ6	7	4.87E+00	1.23E-01	4.72E+00	7.10E+00	4.56E+00	1.17E-01	4.39E+00	6.50E+00
DTLZ7	3	9.95E-03	2.86E-03	4.28E-03	6.70E+00	3.79E-01	1.05E-01	2.18E-01	6.20E+00
DTLZ7	5	2.34E+00	4.31E-01	1.74E+00	6.10E+00	2.76E+00	6.04E-01	1.99E+00	6.20E+00
DTLZ7	7	2.12E+01	1.81E+00	1.80E+01	8.00E+00	1.01E+01	4.26E+00	2.18E+00	5.40E+00

problems is higher with more objectives, and the number of dimensions tackled by both ADMs is also larger, while the number of evaluations was set similarly for all the problems and numbers of objectives. In this regard, an interesting observation is that R-NSGA-II performed usually better than WASF-GA for 5 and 7 objectives, even when employing a lower number of iterations of ADM/ADM-PSO (ITERs).

From the point of view of ADM-PSO's specific performance, it can be seen from Table 3 that it enabled the comparison between WASF-GA and R-NSGA-II and also allowed to capture certain differences in their search strategies. To be more concrete, WASF-GA obtained a better mean (denoted with a grey background in 11 values out of 21 problems) and a minimum distances (11 out of 21 problems) than R-NSGA-II, although the latter needed a slightly lower number of iterations.

To illustrate the behaviour of ADM-PSO with these two iEMOs, Fig. 2 shows representative examples of trajectories walked by the reference points generated, when guiding WASF-GA (left) and R-NSGA-II (right) to solve DTLZ5 with 3 objectives. It can be observed that ADM-PSO with R-NSGA-II resulted with a more spread out trajectory than WASF-GA, whereas the latter algorithm was more concentrated to the area around the ADM-aspiration point (red square symbol). These reference points are in turn generated according to their precedent Pareto front approximations, which are used by ADM-PSO to constitute contextual information. This is indeed illustrated in Fig. 3, where Pareto front approximations are plotted with regards to the 3 closest reference points to the ADM-aspiration point (of ADM-PSO). Accordingly, Pareto front approximations of R-NSGA-II are scattered, whereas WASF-GA showed more concentrated fronts to the ADM-aspiration point area.

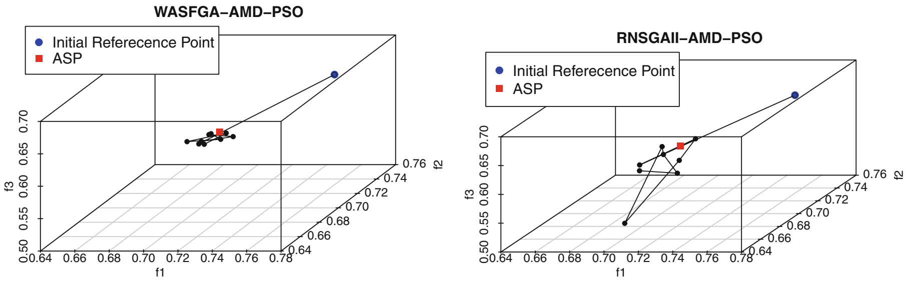

Fig. 2. Search paths of ADM-PSO when guiding WASF-GA (left) and R-NSGA-II (right) to solve DTLZ5 with 3 objectives. (Color figure online)

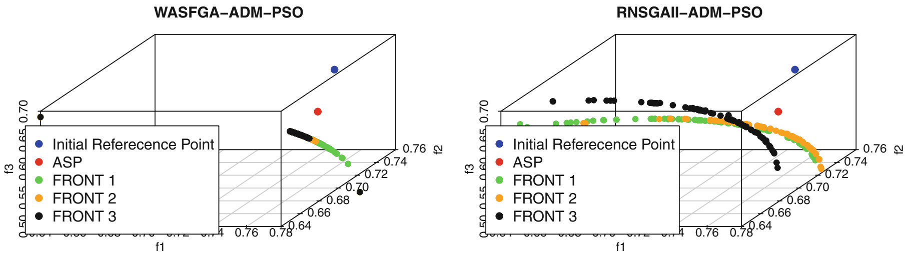

Fig. 3. Pareto front approximations of WASF-GA (left) and R-NSGA-II (right) according to the reference points of ADM-PSO (the 3 closest reference points to the *asp*), when solving DTLZ5 with 3 objectives.

5 Conclusions and Future Work

We have introduced ADM-PSO, a new variant of an ADM for preference articulation in the form of reference points guided by PSO. Our approach enables comparing interactive reference point based EMOs without involving human DMs. ADM-PSO has been implemented following the jMetal architecture and its source codes are freely available.

The proposed approach was demonstrated on the DTLZ benchmark problems with 3, 5 and 7 objectives and using R-NSGA-II and WASF-GA as interactive reference point based methods to be compared. The experimental results show that ADM-PSO is useful and efficient in comparison with the previous ADM. It offers a bio-inspired and flexible mechanism to capture the current context of an ADM in interactive solution processes.

ADM-PSO is conceptually intuitive and straightforward, although it opens a promising line of future research as follows. First, exploring the possibilities of using different metaheuristics like DE, CMA-ES and GA for the generation of reference points instead of PSO. Second, testing parameter tunning in PSO (and other metaheuristics), e.g., φ_1 and φ_2, to control the influence of the current reference point (global best) and/or local history of particles, hence to induce the ADM's behavior in terms of intensification/diversification mechanisms. Third, carrying out further comparisons of multiple state-of-the art iEMOs, to test their performances in a controlled and computationally fair execution framework.

Acknowledgements. This work was partially funded by Grants TIN2017-86049-R (Spanish MICINN) and P12-TIC-1519 (PAIDI). C. Barba-González was supported by Grant BES-2015-072209 (Spanish MICINN) and University of Jyväskylä. J. García-Nieto is the recipient Post-Doct fellowship of "Plan Propio" at Universidad de Málaga. This work was supported on the part of V. Ojalehto by the Academy of Finland (grant number 287496).

References

1. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
2. Miettinen, K., Ruiz, F., Wierzbicki, A.P.: Introduction to multiobjective optimization: interactive approaches. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 27–57. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88908-3_2
3. Miettinen, K., Hakanen, J., Podkopaev, D.: Interactive nonlinear multiobjective optimization methods. In: Greco, S., Ehrgott, M., Figueira, J. (eds.) Multiple Criteria Decision Analysis. ISOR, vol. 233, pp. 931–980. Springer, New York (2016). https://doi.org/10.1007/978-1-4939-3094-4_22
4. López-Ibáñez, M., Knowles, J.: Machine decision makers as a laboratory for interactive EMO. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) EMO 2015. LNCS, vol. 9019, pp. 295–309. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15892-1_20
5. Purshouse, R.C., Deb, K., Mansor, M.M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1147–1154 (2014)
6. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Applications. Wiley, Hoboken (1986)
7. Ojalehto, V., Podkopaev, D., Miettinen, K.: Towards automatic testing of reference point based interactive methods. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.) PPSN 2016. LNCS, vol. 9921, pp. 483–492. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45823-6_45
8. Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM pp. 635–642 (2006)
9. Ruiz, A.B., Luque, M., Miettinen, K., Saborido, R.: An interactive evolutionary multiobjective optimization method: interactive WASF-GA. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) EMO 2015. LNCS, vol. 9019, pp. 249–263. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15892-1_17
10. PSO-Central-Group: Standard PSO 2006, 2007, and 2011. Technical report, Particle Swarm Central, January 2011. http://www.particleswarm.info/
11. Durillo, J.J., Nebro, A.J.: jMetal: a java framework for multi-objective optimization. Adv. Eng. Softw. **42**(10), 760–771 (2011)
12. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. vol. 1, pp. 825–830. IEEE (2002)
13. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, California (2007)