A Multi-objective Particle Swarm Optimizer Based on Decomposition

Saúl Zapotecas Martínez
CINVESTAV-IPN (EVOCINV)
Departamento de Computación
México, D.F. 07360, MEXICO
saul.zapotecas@gmail.com

Carlos A. Coello Coello
CINVESTAV-IPN (EVOCINV)
Departamento de Computación
México, D.F. 07360, MEXICO
ccoello@cs.cinvestav.mx

ABSTRACT

The simplicity and success of particle swarm optimization (PSO) algorithms, has motivated researchers to extend the use of these techniques to the multi-objective optimization field. This paper presents a multi-objective particle swarm optimization (MOPSO) algorithm based on a decomposition approach, which is intended for solving continuous and unconstrained multi-objective optimization problems (MOPs). The proposed decomposition-based multi-objective particle swarm optimizer (dMOPSO), updates the position of each particle using a set of solutions considered as the global best according to the decomposition approach. dMOPSO is mainly characterized by the use of a memory reinitialization process which aims to provide diversity to the swarm. Our proposed approach is compared with respect to two decomposition-based multi-objective evolutionary algorithms (MOEAs) which are representative of the state-of-the-art in the area. Our results indicate that our proposed approach is competitive and it outperforms the two MOEAs with respect to which it was compared in most of the test problems adopted.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search.*

General Terms

Algorithms, Theory.

Keywords

Multi-objective optimization, particle swarm optimization, decomposition approach

1. INTRODUCTION

Particle swarm optimization (PSO) [6] has been found to be a very successful bio-inspired metaheuristic for dealing with unconstrained and continuous optimization problems. This has motivated the interest of researchers in extending PSO to the solution of multi-objective optimization problems (for a survey on multi-objective particle swarm optimizers, see [18]).

Recently, a new multi-objective evolutionary algorithm based on a decomposition approach (MOEA/D) [21] has been proposed. MOEA/D decomposes a MOP into several single-objective optimization problems. In this way, a set of approximate solutions to the Pareto optimal set is reached by minimizing each subproblem instead of using Pareto ranking. This has given rise to a new generation of MOEAs.

Regarding multi-objective particle swarm optimizers (MOPSOs), to the authors' best knowledge, there exist only two of them based on a decomposition approach (see [16, 15]). However, we believe that the potential of this sort of approach, when combined with PSO, has not been fully exploited, which has motivated the work reported in this paper.

Peng and Zhang [16] proposed the multi-objective particle swarm optimizer based on decomposition (MOPSO/D). This approach uses the framework adopted by MOEA/D but replaces the genetic operators (crossover and mutation) by the inertia flight equations used in traditional PSO. MOPSO/D uses a turbulence (or mutation) operator and adopts an archiving strategy (which is based on ϵ-dominance [13]) to store the nondominated solutions found during the search.

More recently, Moubayed et al. [15] proposed a novel smart multi-objective particle swarm optimizer using decomposition (SDMOPSO). This algorithm is also based on MOEA/D, and adopts an external archive based on ϵ-dominance for maintaining diversity in the swarm. An interesting aspect of this approach is that it uses Pareto dominance for updating the personal best position of each particle. In SDMOPSO, the global best set of each particle is defined by all the solutions located within a certain neighborhood, and this set is updated analogously to the mechanism adopted by a MOEA/D variant proposed in [22].

In this paper, we introduce a new decomposition-based multi-objective particle swarm optimizer (dMOPSO) for continuous and unconstrained MOPs. The proposed approach does not require either the use of turbulence or the Pareto optimality concept for approximating solutions towards the Pareto optimal set. Instead, we adopt both a mechanism for selecting global best solutions based on the nature of the decomposition approach and a mechanism to reinitialize the particles based on their age. dMOPSO avoids the use of

GECCO'11, July 12–16, 2011, Dublin, Ireland.

an external archive, which results in a lower computational time than that of state-of-the-art MOPSOs.

The remainder of the paper is organized as follows. In Section 2, we provide the basic background required for understanding the rest of the paper. In Section 3, we explain in detail our proposed approach. In Section 4, we show the results of our comparative study. Finally, in Section 5 we provide our conclusions and some possible paths for future work.

2. BASIC BACKGROUND

2.1 Multi-Objective Optimization

A continuous and unconstrained multi-objective optimization problem can be stated [1] as follows:

$$\min_{x \in \Omega} \quad \{F(x)\} \tag{1}$$

where Ω define the decision variable space and F is defined as the vector of the objective functions:

$$F : \Omega \to \mathbb{R}^k, \quad F(x) = (f_1(x), \ldots, f_k(x))^T$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a continuous and unconstrained function.

In multi-objective optimization, a set of trade-off solutions are normally aimed for, because to minimize a function f_i implies to deteriorate another one. To describe the concept of optimality in which we are interested, the following definitions are provided.

Definition 1. Let $x, y \in \Omega$, we say that x *dominates* y (denoted by $x \prec y$) if and only if, $f_i(x) \leq f_i(y)$ and $F(x) \neq F(y)$.

Definition 2. Let $x^\star \in \Omega$, we say that $x^\star$ is a *Pareto optimal* solution, if there is no other solution $y \in \Omega$ such that $y \prec x^\star$.

Definition 3. The *Pareto Optimal Set* $\mathcal{PS}$ is defined by:

$$\mathcal{PS} = \{x \in \Omega | x \text{ is Pareto optimal solution}\}$$

and its image ($\mathcal{PF} = \{F(x) | x \in \mathcal{PS}\}$) is called the *Pareto Optimal front*.

We are interested in generating as many (different) elements of the Pareto optimal set as possible, while maintaining a distribution of solutions as uniform as possible along the Pareto front.

2.2 Decomposition Approach

Although Pareto ranking has been the most common approach adopted by MOEAs in the last few years, scalarization functions have been available in the literature for a much longer time (see for example [7, 14, 20]). These methods use a weighted vector (as a search direction) to define a scalar function. In this way and under certain assumptions (e.g., the minimum is unique, the weighting coefficients are positive, etc.), an optimal Pareto solution is achieved by minimizing such a scalarization function. Among these methods, probably the two most widely used are the *Tchebycheff* and the *Weighted Sum* approaches. However, as it has been previously discussed in [3, 21], the approaches based on boundary intersection possess certain advantages over those based on either Tchebycheff or the Weighted Sum. Next, we introduce a decomposition approach based on the boundary intersection, which is the approach adopted in this work.

[1]Without loss of generality, we assume minimization

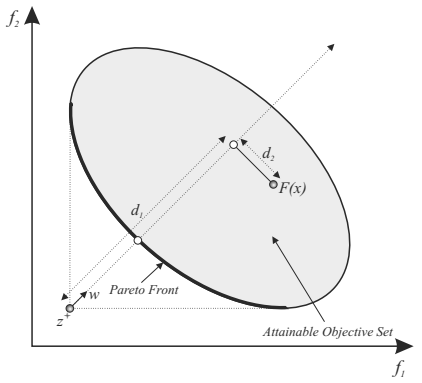


Figure 1: Illustration of the Penalty Boundary Intersection (PBI) approach

2.2.1 Penalty Boundary Intersection Approach

The Penalty Boundary Intersection (PBI)[2] approach proposed by Zhang and Li [21], uses a weighted vector w and a penalty value θ for minimizing both the distance to the utopia vector d_1 and the direction error to the weighted vector d_2 from the solution $F(x)$ (see Fig. 1). Therefore, the optimization problem can be stated as:

$$\text{minimize:} \quad g(x|w, z^\star) = d_1 + \theta d_2 \tag{2}$$

where

$$d_1 = \frac{||(F(x) - z^\star)^T w||}{||w||}$$
$$\text{and} \quad d_2 = \left|\left|(F(x) - z^\star) - d_1 \frac{w}{||w||}\right|\right|$$

as the Tchebycheff approach, $x \in \mathbb{R}^n$ and $z^\star = \min\{f_i(x) | x \in \Omega\}$.

In this way, both the Tchebycheff and the PBI approaches can generate a good representation of the Pareto front by defining a well-distributed set of weighted vectors.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a metaheuristic originally proposed by Kennedy and Eberhart [11] for dealing with continuous and unconstrained nonlinear optimization problems. PSO simulates the movements of a flock of birds which aim to find food. In PSO, a particle is represented by a solution $x_i \in \mathbb{R}^n$. A population can be defined by one or several swarms of particles. Each swarm is evolved by updating both the velocity v^{t+1} and the position of each particle x^{t+1} according to the following equations:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (x_{pb,i} - x_t^t) + c_2 r_2 (x_{gb,i} - x_i^t) \tag{3}$$

and the new particle's position is updated according to the equation:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4}$$

where $w \geq 0$ represents the inertia factor, $c1, c2 \geq 0$ are the constraints on the velocity, r_1, r_2 are two random variables having a uniform distribution in the range $(0, 1)$, $v_i, x_{pb,i}$ and $x_{gb,i}$ represent the velocity, the personal best and the global best position for the i^{th} particle, respectively.

[2]based on the well-known Normal Boundary Intersection (NBI) method [3]

3. THE DECOMPOSITION-BASED MOPSO

3.1 General Framework

Our proposed dMOPSO, employs a decomposition-based framework similar to the one adopted by MOEA/D. Therefore, a well-distributed set of weighted vectors W has to be previously defined. Here, we use the same method as in [21], however, other methods can be used, see for example [2]. The boundary-intersection-based techniques possess certain advantages over Tchebycheff-based techniques (which have been previously discussed in [3] and [21]). Because of that, the PBI approach is adopted in our dMOPSO. However, Tchebycheff or another decomposition approach could also be easily coupled to our approach.

At the beginning of the algorithm, a set of N particles $\mathcal{P} = \{x_1, \ldots, x_N\}$ is randomly initialized. Each particle possesses a flight velocity v_i and an age a_i, both of which are initially set to zero. Along the flight cycles, each particle tries to minimize one of the subproblems defined by the weighted vector w_i. Therefore, each particle undertakes its flight towards a better position in order to minimize a single subproblem $g(x_i|w_i, z^\star)$.

The *personal best* $x_{pb,i}$ of the i^{th} particle, represents the best position of the particle to the i^{th} subproblem. Since at the beginning a particle does not have a previous movement, the best personal position is initialized with the same position as the particle, i.e., $x_{pb,i} = x_i$.

At each cycle, the flight historical record of each particle is used to find the best solutions to each subproblem. Therefore, the set of *global best* $\mathcal{G}_{best}$ is defined in a natural way. This set will contain the solutions that minimize each subproblem, and it is updated in each cycle according to the Algorithm 1. Thus, the notion of elitism used in evolutionary multi-objective optimization is implicitly employed. However, in this case, a decomposition approach is used instead of the more traditional Pareto optimality.

Once the global best set has been defined, the velocity and the position of each particle are updated according to equations (3) and (4), respectively. Since the proposed approach tries to minimize a set of subproblems (whose solutions at the end of the flight cycles should be very close to the Pareto optimal set), and the Algorithm 1 introduces a high selection pressure that should contribute to this, we assume that all solutions in $\mathcal{G}_{best}$ are equally good (i.e., we assume that all the subproblems were satisfactorily solved). Thus, the velocity of each particle is computed using as their global best a solution which is randomly taken from $\mathcal{G}_{best}$.

The age of each particle, promotes the diversity along the flight cycles and indicates when a particle is not providing good information in its flight experience. If a particle does not improve its personal position in a flight cycle, then the particle increases (by one) its age. On the other hand, if a particle exceeds a certain (pre-defined) age threshold,

Algorithm 1 $\mathcal{G}_{best} := updateGlobalBest(W, \mathcal{P})$

1: $\mathcal{T} = \mathcal{P}$, $\mathcal{G} = \emptyset$;
2: **for all** $w_i \in W$ **do**
3: $\quad \mathcal{G} = \mathcal{G} \cup \{x_j | \min_{x_j \in \mathcal{T}} g(x_j|w_i, z^\star)\}$;
4: $\quad \mathcal{T} = \mathcal{T} \setminus \{x_j\}$;
5: **end for**
6: **return** $\mathcal{G}$;

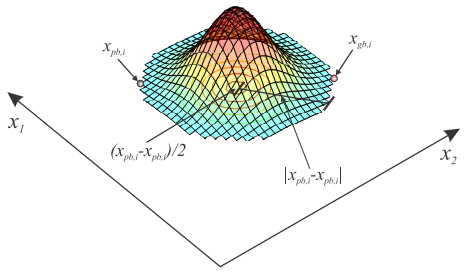


Figure 2: Illustration of the Gaussian distribution used by dMOPSO for generating a new particle. The personal best $x_{pb,i}$ and the global best $x_{gb,i}$ are used to define the *mean* and σ for the normal distribution.

the particle (including, its velocity, its age and its best personal) is reinitialized. dMOPSO employs a reinitialization mechanism based on a parametric probability density function, which involves the selected global best $x_{gb,i}$ and the personal best $x_{bp,i}$ of the current particle x_i^t. With that, dMOPSO aims to perform smart reinitialization movements from the personal best towards the global best solutions using a Gaussian distribution, as is shown in Fig. 2. Therefore, the j^{th} component of the new particle is reset according to the following equation [3]:

$$x_i^{t+1}(j) = N\left(\frac{x_{gb,i}(j) - x_{pb,i}(j)}{2}, |x_{gb,i}(j) - x_{pb,i}(j)|\right) \quad (5)$$

where $N(m, \sigma)$ represents a random number using a normal distribution with mean m and sigma σ.

In PSO, it is very common that some particles go beyond the boundaries of the decision variables. In order to deal with this problem, we used one of the nine methods studied in [1]. Here, we adopted the *deterministic back* to repair solutions that are generated outside the allowable bounds. Therefore, the j^{th} bound of the particle's position x_i^{t+1} and of its velocity v_i^{t+1}, are re-established as follows:

$$\begin{aligned} x_i^{t+1}(j) &= \begin{cases} x_{lb}(j), & \text{if } x_i^{t+1}(j) < x_{lb,i}(j) \\ x_{ub}(j), & \text{if } x_i^{t+1}(j) > x_{ub,i}(j) \end{cases} \\ v_i^{t+1}(j) &= -\gamma v_i^{t+1}(j) \end{aligned} \quad (6)$$

where $x_{lb,i}(j)$ and $x_{ub,i}(j)$ are the lower and upper bounds in the j^{th} component of the allowable decision variable values, respectively. In the same way than other MOPSOs [18, 15], dMOPSO adopts $\gamma = 1$.

Summarizing, the dMOPSO algorithm can be stated as follows:

Step 1) Initialization

Step 1.1) $t = 0$ // the number of flight cycles

Step 1.2) Generate a well-distributed set of N weighted vectors $W = \{w_1, \ldots, w_N\}$.

Step 1.3) Generate a swarm $\mathcal{P}^t = \{x_1 \ldots, x_N\}$ of N random particles.

Step 1.4) Initially, the velocity v_i and the age a_i of each particle x_i is set in zero, i.e. $v_i^t = a_i^t = 0$ (for $i = 1, \ldots, N$).

Step 1.5) Define the personal best: $x_{pb,i} = x_i$, (for $i = 1, \ldots, N$).

[3]This equation is a generalization of the one given in [10].

Step 1.6) Define the global best set: $\mathcal{G}_{best} = \mathcal{P}^t$.

Step 2) Shuffle $\mathcal{G}_{best}$.

Step 3) The cycle // *update* $\mathcal{P}^{t+1} = \{x_1^{t+1}, \ldots, x_N^{t+1}\}$

For $i = 1, \ldots, N$, do

If $a_i^t < T_a$ then

3.1) Update particle: Update the velocity v_i^{t+1} (using as global best the solution $x_{gb,i} \in \mathcal{G}_{best}$) and update the position x_i^{t+1} according to equations (3) and (4), respectively.

Otherwise,

3.2) Reset particle: Reset the memory of the particle, i.e., $v_i^{t+1} = a_i^{t+1} = 0$ and the position x_i^{t+1} is reset according to the equation (5).

3.3) Repair bounds: Repair the particle's bounds x_i^{t+1} and update the velocity v_i^{t+1} according to the equation (6).

3.4) Evaluate the particle and update $z^\star$: Get $F(x_i^{t+1})$. If $f_j(x_i^{t+1}) < z_j^\star$ then $z_j^\star = f_j(x_i^{t+1})$ (for $j = 1, \ldots, k$).

3.5) Update the personal best: If $g(x_i^{t+1}|w_i, z^\star) \leq g(x_{pb,i}|w_i, z^\star)$ then $x_{pb,i} = x_i^{t+1}$ and $a_i^{t+1} = 0$, Else $a_i^{t+1} = a_i^t + 1$.

Step 4) Update the global best: Get $\mathcal{G}_{best}$ from $\mathcal{S} = \mathcal{G}_{best} \cup \mathcal{P}^{t+1}$, i.e., $\mathcal{G}_{best} = updateGlobalBest(W, \mathcal{S})$.

Step 5) Stopping Criteria: If $t < N_{gen}$ then $t = t + 1$ and go to **Step 2**. Otherwise, stop dMOPSO and **output:** $\mathcal{G}_{best}$.

3.2 dMOPSO vs. other decomposition-based MOPSOs

As indicated before, we are only aware of two MOPSOs based on a decomposition approach (see [16, 15]). Next, we compare our proposed MOPSO to them.

MOPSO/D [16], explores the way of obtaining a set of global best solutions using a set of T neighboring solutions. The neighborhood of each particle is updated using a mechanism similar to the one adopted in MOEA/D. Therefore, for a swarm of size N, the computational complexity of MOPSO/D is $\mathcal{O}(NT)$. On the other hand, SDMOPSO [15] also uses a neighborhood of size T as its global best set. However, only n_r (where $n_r < T$) solutions of the neighborhood are updated. Nonetheless, the worst case occurs when T solutions are compared to obtained those n_r solutions for updating. Considering N particles in the swarm, the computational complexity for this updating process is as well given by $\mathcal{O}(NT)$. Both MOPSO/D and SDMOPSO use an external archive. When such an external file is completely full, the computational effort required to decide whether a new particle is stored in the archive is $\mathcal{O}(N)$. Considering an archive size of N, the updating process has then a computational complexity of $\mathcal{O}(N^2)$. Therefore, in each cycle, both MOPSO/D and SDMOPSO require a computational complexity of $\mathcal{O}(NT + N^2)$. In dMOPSO, the global best set is only updated once at each cycle, i.e., we find the best solution for each subproblem of the decomposition approach. This updating has a computational complexity of $\mathcal{O}(N^2)$ for N different subproblems (as shown in Algorithm 1). In fact, when the number of generations is large, the computational cost offered by dMOPSO become significantly lower than either MOPSO/D or SDMOPSO.

dMOPSO does not use any explicit diversity maintenance mechanism to obtain well-distributed solutions along the Pareto front. Instead, it relies on the penalty from the PBI approach, which provides diversity to such solutions along the flight circuits. However, the use of other decomposition approaches could deteriorate the performance of our dMOPSO, especially when dealing with more complex MOPs. This is because the PBI approach forces particles to follow a single direction, since moving away from such direction will be penalized. dMOPSO does not use any archiving strategy and, instead, it relies on the fact that at the end of flight cycles the subproblems will be minimized and, thereby, nondominated solutions will be reached. Nevertheless, the use of archiving strategies (see e.g. [12, 13, 24]) could be used to cover those drawbacks that weighted-vector-based methods possess (e.g. when dealing with MOPs having either disconnected Pareto fronts or many objective functions). They could be employed to store well-distributed nondominated solutions, while the set of global best solutions could be used as guides in the optimization process.

4. EXPERIMENTAL RESULTS

4.1 Test Problems

In order to assess the performance of the proposed approach, we compare its results with respect to those obtained by a MOEA/D [21] and SDMOPSO [15]. We adopted nine test problems whose Pareto fronts have different characteristics including convexity, concavity, disconnections and multi-frontality. The first adopted MOP was proposed by Fonseca [8]. The two-objective test suite of Zitzler-Deb-Thiele (ZDT) [23] (except for ZDT5, which is a binary problem) is also adopted. For three-objective problems, we adopted three problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [5] (DTLZ2, DTLZ6 and DTLZ7).

We used 30 decision variables for ZDT1, ZDT2 and ZDT3. ZDT4 and ZDT6 was tested using 10 decision variables. For DTLZ2 and DTLZ6, 12 variables were adopted. DTLZ7 was tested with 22 decision variables. Finally, Foseca's problem was tested using only three decision variables.

4.2 Performance Measures

4.2.1 Hypervolume

The *Hypervolume* ($\mathcal{H}v$) performance measure was proposed by Zitzler [25]. This performance measure is Pareto compliant [26] and quantifies the approximation of nondominated solutions to the Pareto optimal front. The hypervolume corresponds to the non-overlapped volume of all the

Table 1: Parameters for dMOPSO, SDMOPSO and MOEA/D

Parameter	dMOPSO	SDMOPSO	MOEA/D
N_{pop}	100/300	100/300	100/300
N_{gen}	150	150	150
N_{arc}	–	100/300	–
ϵ	–	0.0075	–
n_r	–	2	–
T_n	–	30	30
η_c	–	–	20
η_m	–	–	20
P_c	–	–	1
P_m	–	–	1/n
T_a	2	–	–
θ	5	5	5

Table 2: Results of the $\mathcal{H}v$ performance measure for dMOPSO, SDMOPSO and MOEA/D

MOP	dMOPSO	SDMOPSO	MOEA/D
	average (σ)	*average* (σ)	*average* (σ)
FONSECA	**0.546328** (0.000137)	0.541803 (0.001744)	0.383232 (0.001008)
ZDT1	**0.869828** (0.000160)	0.102649 (0.064542)	0.809607 (0.030636)
ZDT2	**0.536473** (0.000425)	0.026707 (0.035759)	0.316948 (0.091569)
ZDT3	**1.317248** (0.002921)	0.151858 (0.090791)	1.246748 (0.044609)
ZDT4	**0.862245** (0.029966)	0.000000 (0.000000)	0.774609 (0.065285)
ZDT6	**0.504519** (0.000004)	0.052415 (0.084859)	0.457862 (0.009460)
DTLZ2	0.741219 (0.002131)	0.713603 (0.005012)	**0.771041** (0.000386)
DTLZ6	**0.426532** (0.000038)	0.332548 (0.149801)	0.426153 (0.000098)
DTLZ7	**1.409133** (0.007166)	1.115480 (0.068608)	1.375630 (0.141559)

hypercubes formed by a reference point r (given by the user) and each solution p in the Pareto set approximation ($\mathcal{PF}_k$). Hypervolume is mathematically defined as follows:

$$\mathcal{H}v = \Lambda\left(\bigcup_{p\in\mathcal{PF}_k}\{x|p \prec x \prec r\}\right) \quad (7)$$

where Λ denotes the Lebesgue measure and $r \in \mathbb{R}^k$ denotes a reference vector being dominated by all valid candidate solutions in PF_k.

4.2.2 Spacing

The *Spacing* ($\mathcal{S}p$) performance measure was proposed by Schott [19], and quantifies the spread of solutions in the obtained approximation of the Pareto front. This performance measure is computed as:

$$\mathcal{S}p = \sqrt{\frac{1}{|P|}\sum_{i=1}^{|P|}(\overline{d} - d_i)^2} \quad (8)$$

where d_i and $\overline{d}$ are defined as: $d_i = \min_{i,i\neq j}\sum_{k=1}^{M}|f_k^i - f_k^j|$ and $\overline{d} = \frac{\sum_{i=1}^{|P|} d_i}{|P|}$. A value of zero for this performance measure indicates that all the solutions are uniformly spread (i.e., the best possible performance).

4.2.3 Coverage of Two Sets

The *Coverage of Two Sets* ($\mathcal{C}$) was proposed by Zitzler et al. [23], and it compares two sets of solutions A and B, using Pareto dominance. This performance measure is defined as:

$$\mathcal{C}(A,B) = \frac{|\{b \in B|\exists a \in A : a \preceq b\}|}{|B|} \quad (9)$$

If all points in A dominate or are equal to all points in B, this implies that $\mathcal{C}(A,B) = 1$. Otherwise, if no point of A dominates some point in B then $\mathcal{C}(A,B) = 0$. When $\mathcal{C}(A,B) = 1$ and $\mathcal{C}(B,A) = 0$ then, we say that A is better than B. Since the Pareto dominance relation is not symmetric, we need to compute both $\mathcal{C}(A,B)$ and $\mathcal{C}(B,A)$.

Table 3: Results of the $\mathcal{S}p$ performance measure for dMOPSO, SDMOPSO and MOEA/D

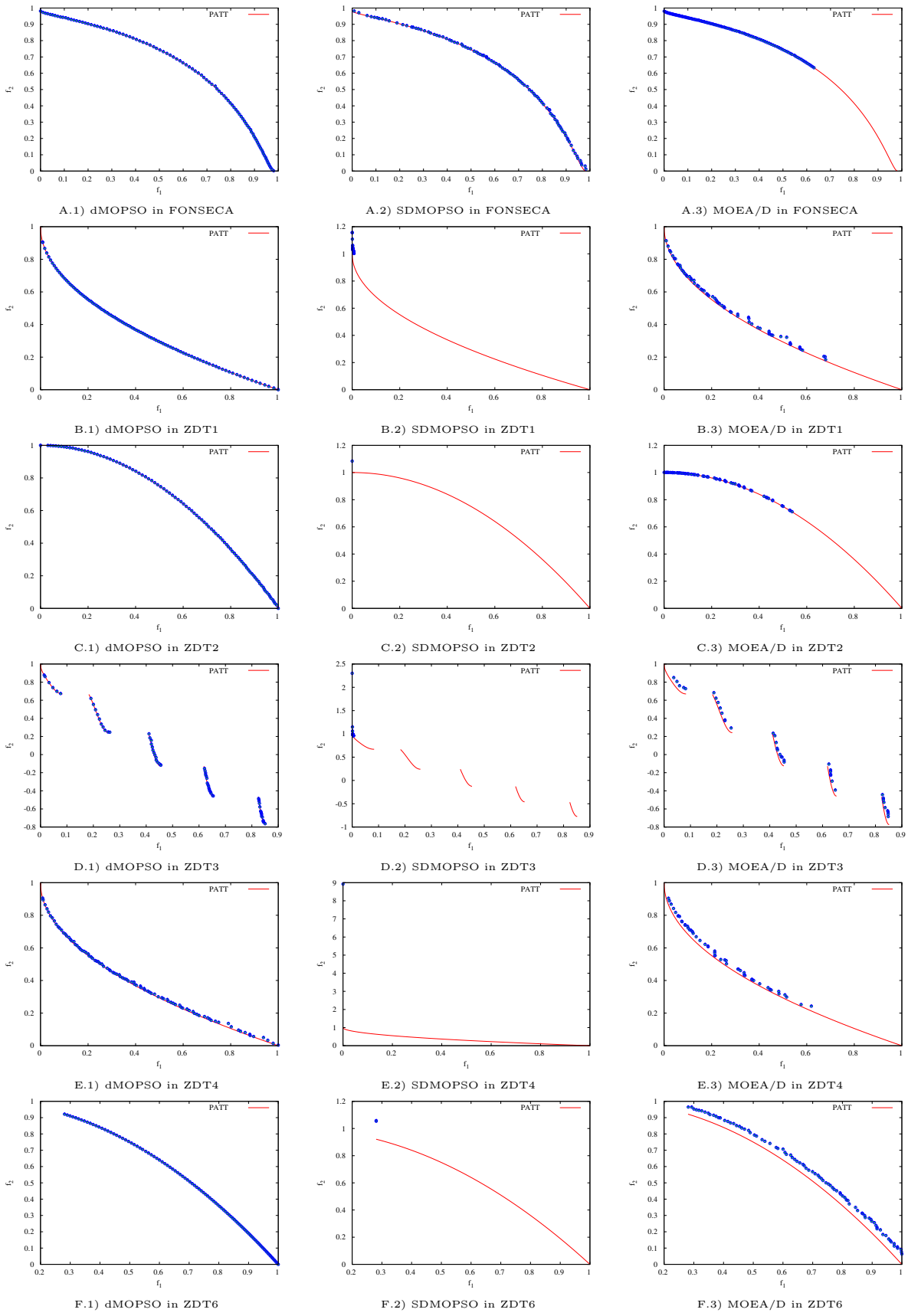
MOP	dMOPSO	SDMOPSO	MOEA/D
	average (σ)	*average* (σ)	*average* (σ)
FONSECA	0.004118 (0.000104)	0.004597 (0.000739)	**0.003055** (0.000074)
ZDT1	**0.004822** (0.000157)	0.032203 (0.037486)	0.007400 (0.002690)
ZDT2	0.004231 (0.000178)	**0.000587** (0.002528)	0.008435 (0.007557)
ZDT3	**0.016659** (0.000752)	0.022242 (0.041666)	0.019219 (0.003819)
ZDT4	**0.006004** (0.005617)	0.020263 (0.048561)	0.009245 (0.003852)
ZDT6	**0.002762** (0.000014)	0.421809 (0.845177)	0.005847 (0.001834)
DTLZ2	**0.024236** (0.001323)	0.030467 (0.002005)	0.029092 (0.000179)
DTLZ6	**0.003563** (0.000366)	0.581048 (1.036406)	0.036316 (0.046853)
DTLZ7	0.103306 (0.001991)	**0.040643** (0.010550)	0.100617 (0.021034)

Table 4: Results of the $\mathcal{C}$ performance measure for dMOPSO, SDMOPSO and MOEA/D

MOP	$\mathcal{C}$(dMOPSO, SDMOPSO)	$\mathcal{C}$(SDMOPSO, dMOPSO)	$\mathcal{C}$(dMOPSO, MOEA/D)	$\mathcal{C}$(MOEA/D, dMOPSO)
	average (σ)	*average* (σ)	*average* (σ)	*average* (σ)
FON	**0.226667** (0.034383)	0.034333 (0.016059)	0.026333 (0.018883)	**0.090667** (0.019653)
ZDT1	**0.028328** (0.093794)	0.000000 (0.000000)	**0.895667** (0.089132)	0.000000 (0.000000)
ZDT2	**1.000000** (0.000000)	0.000000 (0.000000)	**0.499667** (0.172153)	0.048667 (0.028016)
ZDT3	**0.103268** (0.207493)	0.001000 (0.005385)	**0.815667** (0.101937)	0.229000 (0.095231)
ZDT4	**0.256550** (0.425500)	0.000000 (0.000000)	**0.989333** (0.015041)	0.000000 (0.000000)
ZDT6	**1.000000** (0.000000)	0.000000 (0.000000)	**1.000000** (0.000000)	0.000000 (0.000000)
DTLZ2	**0.311188** (0.066503)	0.175778 (0.029312)	0.000556 (0.001242)	**0.645222** (0.009988)
DTLZ6	**0.299783** (0.392221)	0.296222 (0.156012)	**0.477556** (0.022326)	0.368333 (0.008378)
DTLZ7	**0.577711** (0.223492)	0.041667 (0.013465)	0.221444 (0.059589)	**0.427222** (0.102300)

4.3 Experimental Settings

For each MOP, 30 independent runs were performed with each algorithm. The parameters used in each algorithm are summarized in Table 1, where N_{pop} represents the population size (100 for the bi-objective and 300 for the three-objective problems). N_{gen} represents the number of generations (which was set in 150). Therefore, we performed 15,000 (for the bi-objective problems) and 45,000 (for the three-objective problems) fitness function evaluations. For dMOPSO, T_a represents the age threshold. For SDMOPSO, N_{arc} represents the maximum number of elements in the external archive, ϵ is the parameter used for ϵ-dominance, n_r is the number solutions which are replaced in the neighborhood and T_n defines the neighborhood size. For MOEA/D, T_n, η_c, η_m, P_c and P_m represent the neighborhood size, crossover index, mutation index, crossover rate and mutation rate, respectively. Finally, the parameter θ, represents the penalty value used in the PBI approach. For dMOPSO and SDMOPSO, the constraints on the velocity (c_1, c_2) and the inertia factor (w) were dynamically defined (as in other



A.1) dMOPSO in FONSECA A.2) SDMOPSO in FONSECA A.3) MOEA/D in FONSECA

B.1) dMOPSO in ZDT1 B.2) SDMOPSO in ZDT1 B.3) MOEA/D in ZDT1

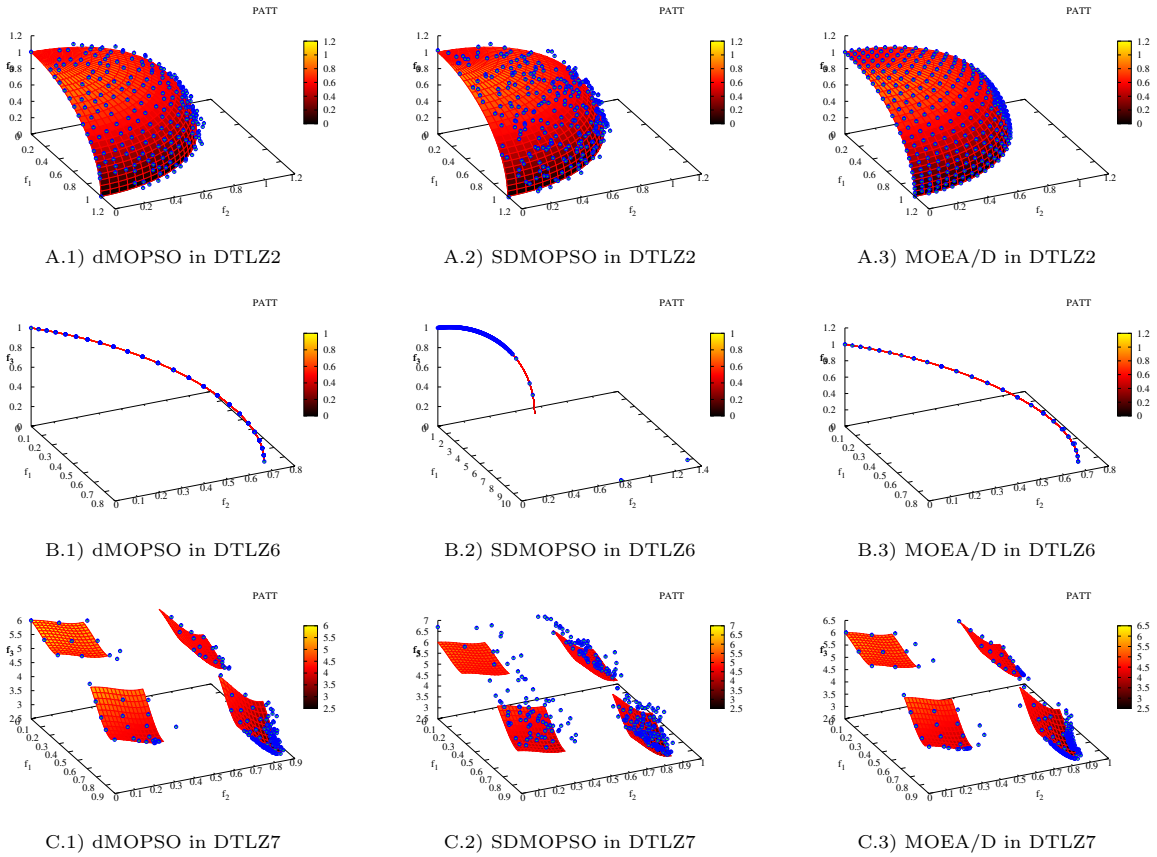
C.1) dMOPSO in ZDT2 C.2) SDMOPSO in ZDT2 C.3) MOEA/D in ZDT2

D.1) dMOPSO in ZDT3 D.2) SDMOPSO in ZDT3 D.3) MOEA/D in ZDT3

E.1) dMOPSO in ZDT4 E.2) SDMOPSO in ZDT4 E.3) MOEA/D in ZDT4

F.1) dMOPSO in ZDT6 F.2) SDMOPSO in ZDT6 F.3) MOEA/D in ZDT6

Figure 3: dMOPSO, SDMOPSO and MOEA/D in the bi-objective optimization problems



A.1) dMOPSO in DTLZ2 A.2) SDMOPSO in DTLZ2 A.3) MOEA/D in DTLZ2

B.1) dMOPSO in DTLZ6 B.2) SDMOPSO in DTLZ6 B.3) MOEA/D in DTLZ6

C.1) dMOPSO in DTLZ7 C.2) SDMOPSO in DTLZ7 C.3) MOEA/D in DTLZ7

Figure 4: dMOPSO, SDMOPSO and MOEA/D in the three-objective optimization problems

MOPSOs, see e.g. [17, 15]). Additionally, we used the values proposed by the authors of SDMOPSO [15], which take uniformly distributed values, such that: $c_1, c_2 \in (1.2, 2.0)$ and $w \in (0.1, 0.5)$.

For each MOP, the algorithms were evaluated using the three performance measures previously indicated (*Hypervolume, Spacing* and *Coverage of Two Sets*). The results are summarized in Tables 2 to 4. Regarding the hypervolume, the reference vectors adopted were: $r = (1.1, \ldots, 1.1)^T$ for FONSECA, DTLZ2, DTLZ6 and ZDT test problems, while for DTLZ7 the reference vector $r = (1.0, 1.0, 6.1)^T$ was used.

Each table displays both the *average* and the standard deviation (σ) of each performance measure, for each of the test problems adopted. The best results are shown in **boldface** for each performance measure and test problem adopted.

4.4 Discussion of Results

As shown in Table 2, our proposed approach (dMOPSO) outperformed both SDMOPSO and MOEA/D in most of the test problems with respect to the *Hypervolume* ($\mathcal{H}v$). This indicates that our algorithm produced a better approximation along the Pareto front. The exception was DTLZ2, where MOEA/D obtained better results. However, the differences in the values obtained by our approach are not significant.

Regarding the *Spacing* ($\mathcal{S}p$) performance measure, our proposed dMOPSO obtained better results with respect to those produced by SDMOPSO and MOEA/D in most test problems. The exceptions were ZDT2, DTLZ7 and FONSECA. However, a better distribution of solutions is relevant only when there is a good approximation to the Pareto front. For these specific problems (FONSECA, DTLZ2 and DTLZ7), our proposed approach achieved better convergence than both SDMOPSO and MOEA/D.

Finally, according to the *Coverage of Two Sets* ($\mathcal{C}$) performance measure, dMOPSO outperformed the results obtained by SDMOPSO in all the test problems adopted. That means that our proposed approach obtained more solutions that dominate those generated by SDMOPSO. However, for the FONSECA, DTLZ2 and DTLZ7 test problems MOEA/D obtained the better results. However, in the case of FONSECA, it can be clearly seen in Fig. 3 that MOEA/D only covered a portion of the true Pareto front, whereas dMOPSO covered it entirely. So, in this case, MOEA/D generated more nondominated solutions than dMOPSO, but they were not properly distributed. Something similar happens in DTLZ7, being DTLZ2 the only case in which it is clear that MOEA/D outperformed our dMOPSO, although the results (both numerical and graphical values) regarding convergence and spread are not significantly different.

Figures 3 and 4 show the final Pareto fronts obtained by dMOPSO, SDMOPSO and MOEA/D algorithms in the adopted test problems. These plots show the final set of nondominated solutions found by each algorithm and correspond to the run with the value nearest to average of the $\mathcal{H}v$ metric for each problem.

5. CONCLUSIONS AND FUTURE WORK

We have presented a multi-objective particle swarm optimizer based on decomposition. Our proposed algorithm was able to outperform both MOEA/D and SDMOPSO in most of the test problems adopted, with respect to three performance measures (hypervolume, spacing and coverage of two sets). Our proposed approach does not use an external archive to store nondominated solutions and adopts instead, a mechanism to select the globally best solutions.

The use of PBI greatly benefits the diversity of solutions in the minimization process of our approach. The penalty parameter θ imposes a search direction defined by each weighted vector. However, a study of the variation of this parameter along the flight circuits is a promising path for future research. Evidently, it could relax the search process, and could lead us to achieve faster convergence towards the Pareto optimal set.

As part of our future work, we plan to exploit the properties of the decomposition approach when coupling it to other metaheuristics (e.g., scatter search [9] or artificial immune systems [4]) in order to design a more robust strategy which can effectively deal with more complicated multi-objective problems.

Acknowledgments

The first author acknowledges support from CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 103570.

6. REFERENCES

[1] M. Clerc. Confinements and Biases in Particle Swarm Optimisation, 2006.

[2] I. Das. *Nonlinear Multicriteria Optimization and Robust Optimality*. PhD thesis, Rice University, Houston, Texas, 1997.

[3] I. Das and J. E. Dennis. Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[4] L. N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[6] R. C. Eberhart, J. Kennedy, and Y. Shi. *Swarm Intelligence*. Morgan Kaufmann, 1st edition, April 2001.

[7] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition edition, June 2005.

[8] C. M. Fonseca and P. J. Fleming. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 42–52, Sheffield, UK, September 1995. IEE.

[9] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39:653–684, 2000.

[10] J. Kennedy. Bare bones particle swarms. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 80–87, April 2003.

[11] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[12] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[13] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb. Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization. In W. L. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 439–447, San Francisco, California, July 2002. Morgan Kaufmann Publishers.

[14] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachuisetts, 1999.

[15] N. A. Moubayed, A. Petrovski, and J. A. W. McCall. A novel smart multi-objective particle swarm optimisation using decomposition. In *PPSN (2)*, pages 1–10, 2010.

[16] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *IEEE International Conference on Granular Computing, 2008. GrC 2008*, pages 534 –537, 2008.

[17] M. Reyes Sierra and C. A. Coello Coello. Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ-Dominance. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 505–519, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[18] M. Reyes-Sierra and C. A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[19] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[20] P. Vincke. *Multicriteria Decision-Aid*. John Wiley & Sons, New York, 1992.

[21] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[22] Q. Zhang, W. Liu, and H. Li. The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 203–208, Trondheim, Norway, May 2009. IEEE Press.

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

[24] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[25] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[26] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.