MOPSOhv: A New Hypervolume-based Multi-Objective Particle Swarm Optimizer

Ivan Chaman García[1], Carlos A. Coello Coello[1] and Alfredo Arias-Montaño[2]
[1]CINVESTAV-IPN, Av. IPN 2508, San Pedro Zacatenco, Gustavo A. Madero, México D.F. 07360, MEXICO
e-mail: chaman030687@gmail.com ccoello@cs.cinvestav.mx
[2]Instituto Politécnico Nacional, ESIME Ticoman, Av. Ticoman No. 600, Col. San José Ticoman, México, D. F., 07340, MEXICO
e-mail: aarias@ipn.mx

***Abstract*—This paper proposes a new hypervolume-based multi-objective particle swarm optimizer (called MOPSOhv) that uses an external archive to store the global nondominated solutions found during the evolutionary process. The proposed algorithm makes use of the hypervolume contribution of archived solutions for selecting global and personal leaders for each particle in the main swarm, and also as a mechanism for pruning the external archive when it is updated with new nondominated solutions. In order to increase the diversity when particles are updated in their positions, a mutation operator is used. The performance of the proposed algorithm is evaluated adopting standard test problems and indicators reported in the specialized literature, comparing its results with respect to those obtained by state-of-the-art multi-objective evolutionary algorithms. Our preliminary results indicate that our proposal is competitive with respect to state-of-the-art multi-objective evolutionary algorithms, being particularly suitable for solving many-objective optimization problems (i.e., problems having more than 3 objectives).**

I. Introduction

There are many industrial and engineering problems, whose solution, require to simultaneously optimize several objectives which are in conflict with each other, i.e., the improvement of one objective implies the deterioration of another in the problem. These are the so-called *Multi-Objective Problems (MOPs).* Multi-Objective Evolutionary Algorithms (MOEAs) have been very successful in solving MOPs, mainly due to their following features [1]: They do not require any specific knowledge about the problem, they can be used as effective and robust global optimizers, they are easy to understand and implement both in sequential and parallel platforms and also, they can be hybridized with mathematical programming techniques and even with other metaheuristics. Additionally, MOEAs are less susceptible to the shape and continuity of the Pareto front than mathematical programming methods [2].

The use of MOEAs is particularly suitable for solving MOPs since they work simultaneously with a set of potential solutions (i.e., the swarm). This feature allows them to find several solutions of the Pareto optimal set in a single run. Many MOEAs have been developed to solve MOPs such as: PAES [3], MOEA/D [4], SPEA [5], SPEA2 [6], NSGA-II [7] and SMS-EMOA [8], just to name a few.

Particle Swarm Optimization (PSO) is a metaheuristic inspired on the choreography of a bird flock which aims to find food [9]. PSO can be seen as a distributed behavioral algorithm that performs (in its more general version) a multidimensional search. The implementation of the algorithm adopts a population of particles which are initialized with random solutions (random positions in the design space), and whose behavior is affected by either the best local (i.e., within a certain neighborhood) or the best global individual. Over the generations, particles adapt their beliefs to the most successful solutions found in their environment. Each particle has a position and velocity vector that controls its movement, and is updated according to the following general rules:

$$\vec{v}_i^{t+1} \leftarrow \omega \vec{v}_i^{t} + \varphi_1 r_1 \left(\vec{pBest}_i^{t} - \vec{x}_i^{k} \right) + \varphi_2 r_2 \left(\vec{gBest}^{t} - \vec{x}_i^{k} \right) \tag{1}$$
$$\vec{v}_i \in [\vec{v}_{min}, \vec{v}_{max}]$$

$$\vec{x}_i^{t+1} \leftarrow \vec{x}_i^{t} + \vec{v}_i^{t+1} \tag{2}$$
$$x_i \in [x_i^{min}, x_i^{max}]$$

where ω is the velocity inertia factor, φ_1 and φ_2 are the cognitive and social factors respectively, r_1 and r_2 are random numbers, $pBest$ and $gBest$ represent the personal and global leaders. These positions will influence the particle's velocity ($\vec{v}$) and position update. For extending PSO to deal with MOPs, the main issues are the following:

(1) how to select particles (to be used as leaders) in order to give preference to nondominated solutions over those that are dominated?,
(2) how to retain the nondominated solutions found during the search process in order to report solutions that are nondominated with respect to all the past populations and not only with respect to the current one?, and
(3) how to maintain diversity in the swarm in order to avoid convergence to a single solution?

Normally, mechanisms very similar to those adopted with MOEAs (namely, Pareto-based selection and external archives) have been adopted in multi-objective particle swarm optimizers (MOPSOs). However, the addition of other mechanisms (e.g., a mutation operator) is also relatively common in MOPSOs.

Some relevant MOPSOs are the following: In [10] is presented the use of a PSO with a weighted aggregation technique and a fully connected topology as the neighborhood for each particle in the swarm. In [11], [12], PSO is provided with a dynamic neighborhood strategy, particle memory updating, and the process corresponds to a one-dimension optimization, i.e., one objective is optimized at a time. Other MOPSOs make use of several sub-populations or subswarms, to increase the covering of the Pareto front approximation [13], adopt a decomposition method [14], or solve the MOP using a parallel platform [15]. In other approaches the use of a leader selection scheme based on Pareto dominance is adopted [16], [17]. In some other research works, traditional MOEAs' mechanisms for keeping diversity such as the *crowding operator* [18], [19], and *niching* [17] are adopted. Parallel implementation of MOPSOs has been proposed using Grid Computing [15], Master-Slave model [20], and GPUs [19], [21]. For a more thorough review of different MOPSOs see [22]

It is important to remark that most of the previously indicated approaches, incorporate a dynamic scheme, which updates the velocity inertia factor (ω), make use of an external archive and introduce a turbulence factor for increasing the diversity of solutions.[1] In this work, we propose the use of the hypervolume contribution of archived solutions for selecting each swarm particle's global and personal leaders, and also as a mechanism for updating the external archive when inserting new non-dominated solutions into it. These mechanisms had not yet been explored in the context of MOPSO's design to the authors' best knowledge. Also, and in order to increase the diversity of solutions when updating their positions, a mutation operator is adopted.

The remainder of this paper is organized as follows. Section II presents some basic concepts of multi-objective optimization. The proposed algorithm, called MOPSOhv, is described in Section III. Section IV, defines the experimental design used for validating our proposal. Section V reports the peformance obtained for our proposed approach adopting standard performance measures reported in the specialized literature. In this section, we perform a small scalability study to analyze the behavior of our proposed MOPSO when solving many-objective optimization problems (i.e., MOPs with more than 3 objectives). Finally, Section VI provides our conclusions and some possible paths for future research.

II. Basic Concepts

When solving MOPs, the aim is to find a set of decision variable vectors wich represents the best possible trade-offs among all the objetives. The most commonly adopted approach for solving MOPs is to compare solutions (i.e., decision variables vectors) by using the Pareto dominance relation. Some concepts regarding Pareto dominance are briefly described next.

Definition 1. *Dominance: Given two decision variable vectors $\vec{x}, \vec{y} \in R^n$ and a function $F : R^n \to R^k$, $\vec{x}$ dominates $\vec{y}$ ($\vec{x} \preceq \vec{y}$) if and only if*

[1]This is helpful in terms of preventing premature convergence in the presence of local Pareto fronts in MOPs.

$$\forall_i \in \{1, \ldots, k\} \qquad f_i(\vec{x}) \leq f_i(\vec{y})$$
$$\exists_j \in \{1, \ldots, k\} \qquad f_j(\vec{x}) < f_j(\vec{y})$$

otherwise $\vec{x} \npreceq \vec{y}$

Within Pareto dominance, we can distinguish between strong dominance and weak dominance.

Definition 2. *Strong dominance: A solution $\vec{x}$ strongly dominates $\vec{y}$ if $\vec{x}$ is strictly better than $\vec{y}$ in all objectives.*

Definition 3. *Weak dominance: A solution $\vec{x}$ weakly dominatess $\vec{y}$ if $\vec{x}$ is better than $\vec{y}$ in at least one objective and is as good as $\vec{y}$ is all other objectives.*

Neither type of Pareto dominance induces a total order in R^k since some solutions may be *incomparable*. Therefore, MOPs normally do not have a single solution but a set of incomparable solutions which is called the *Pareto optimal set*.

Definition 4. *Pareto optimal set: In a MOP, the Pareto optimal set P is defined as:*

$$P = \left\{\vec{x} \in R^k | \quad \forall \vec{y} \in R^k \vec{y} \npreceq \vec{x}\right\}$$

Definition 5. *Pareto front: Given a MOP and its Pareto optimal set P, the Pareto front is defined as:*

$$PF = \{\vec{u} = (f_1(\vec{x}), \ldots, f_k(\vec{x})) | \vec{x} \in P\}$$

The Pareto front of a MOP is bounded by the *ideal* and *nadir* objective vectors.

Definition 6. *Ideal and nadir vectors: Given a MOP and its the Pareto optimal set P, the ideal objective vector is defined as:*

$$f_{ideal} = \left(\substack{inf \\ \vec{x} \in P} f_1(\vec{x}), \ldots, \substack{inf \\ \vec{x} \in P} f_k(\vec{x})\right)$$

If the ideal objective vector represents an existing solution, then the solution of the MOP is unique.

Analogously, the Nadir objective vector is defined as:

$$f_{nadir} = \left(\substack{sup \\ \vec{x} \in P} f_1(\vec{x}), \ldots, \substack{sup \\ \vec{x} \in P} f_k(\vec{x})\right)$$

III. Our proposed approach

In our proposed approach, which is called *Multi-Objective Particle Swarm Optimizer based on hypervolume* (MOPSOhv), the hypervolume contribution of archived solutions (A^t) is incorporated first, to select the individuals that will be considered as the global and personal leaders for each particle in the swarm, and then, as a mechanism for inserting new non-dominated solutions in the external archive, i.e., particles with the highest hypervolume contribution are kept in the archive A^t, while particles with the least contribution are deleted. We also adopt a mutation operator in order to maintain diversity in the swarm when the position of the particles is updated. This condition is meant to prevent the swarm from prematurely converging to a single solution or to a local Pareto front.

A. The Algorithm

Algorithm 1 presents the pseudocode for our proposed approach.

Algorithm 1: MOPSOhv

```
$N$, Swarm size ;
$G_{max}$, maximum number of generations ;
$P_{MUT}$ mutation probability;
$\omega$ velocity inertia factor;
$\varphi_1$ cognitive factor;
$\varphi_2$ social factor;
$t \leftarrow 0$ ;
Random Initialize $\vec{x}_i^0$ and $\vec{v}_i^0$, $i = 1, 2, \ldots, N$ in swarm ;
Evaluate $f_j(\vec{x}_i^0)$ , $j = 1, 2, \ldots, k$ ; $i = 1, 2, \ldots, N$ ;
$A^0 \leftarrow InitializeExternalArchive()$ ;
while $t < G_{max}$ do
    $HvContribution(A^t)$ ;
    $SortDecreasingHv(A^t)$ ;
    for $i = 1$ to $N$ do
        $gBest_i^t \leftarrow RandomSelect(A^t, TOP)$ ;
        $pBest_i^t \leftarrow RandomSelect(A^t, BOT)$ ;
        $\vec{v}_i^{t+1} \leftarrow UpdateVel(v_i^t, \omega, \varphi_1, \varphi_2, gBest_i^t, pBest_i^t)$ ;
        $\vec{x}_i^{t+1} \leftarrow UpdatePos(x_i^t, v_i^{t+1})$ ;
        $BoundParticle()$ ;
        if $t < (G_{max} \cdot P_{MUT})$ then
            $MutateParticle(x_i^{t+1})$ ;
        end if ;
        Evaluate $f_j(\vec{x}_i^{t+1})$ , $j = 1, 2, \ldots, k$ ;
    end for ;
    for $i = 1$ to $N$ do
        $A^{t+1} \leftarrow UpdateExternalArchiveHv(A^t, x_i^{t+1})$ ;
    end for ;
    $t \leftarrow t + 1$ ;
end while
return Nondominanted solutions in external archive $A^{G_{max}}$ ;
```

The N particles that constitute the swarm are randomly initialized in their positions and velocities and are evaluated in the k objective functions defined in the problem. From the initial swarm, the nondominated solutions are copied and inserted into the initial external archive A^0 (procedure $InitializeExternalArchive()$). The main loop of our proposed approach consists of the following steps: The hypervolume contribution for each particle contained in the external archive is first determined (procedure $HvContribution(A^t)$). Then, these particles are sorted in decreasing order with respect to their hypervolume contribution (procedure $SortDecreasingHv(A^t)$). Once the external archive is sorted, the global ($gBest_i^t$) and personal ($pBest_i^t$) leaders are selected for each particle in the swarm. The global leader is chosen from the top portion particles (i.e., from the top 2%) of the sorted external archive A^t, while the personal leader is chosen from the bottom portion particles (i.e., the bottom 98%). Whith these leaders, each particle in the swarm is updated in its velocity (procedure $UpdateVel()$) and position (procedure $UpdatePos()$). The new particle's position must be bounded with respect to the design space (procedure $BoundParticle()$). For this, if particle x_i^{t+1} goes beyond a boundary in any decision variable, then it is reinserted into the design space by making the decision variable to take the value of its corresponding lower or upper boundary value and its velocity is multiplied by -1 so that it searches in the opposite direction. The final step in updating each particle is to apply a mutation operator, which depends on a certain mutation probability P_{MUT} and the time in the evolutionary process (t). Once all the particles have been updated in position and velocity, they are evaluated in the objective functions defined in the problem, and the new nondominated solutions in the swarm are selected for updating the external archive A^{t+1} (procedure $UpdateExternalArchive()$). In this final step, it might happen that new solutions dominate existing archive solutions, in which case the latter are deleted. However, it might also happen that new solutions are nondominated with respect to the entire external archive, and inserting a new solution will overfill the predefined size of it. In this later case, new nondomited solutions are inserted into the external archive, which is then pruned to its maximum allowable size, by deleting the particles which less contribute to the hypervolume.

B. Hypervolume Contribution

For our proposed MOPSOhv, it is required to compute the hypervolume contribution for each particle in the swarm. We present next the definition of the hypervolume:

Definition 7. *Hypervolume (Hv): Given a Pareto approximation set PF_{known}, and a reference point in objective space z_{ref}, the hypervolume estimates the non-overlaping volume of all the hypercubes formed by the reference point and every vector in the Pareto set approximation. This is mathematically defined as:*

$$HV = \{\cup_i vol_i | vec_i \in PF_{known}\} \quad (3)$$

vec_i is a nondominated vector from the Pareto set approximation, and vol_i is the volume for the hypercube formed by the reference point and the nondominated vector vec_i.

In the context of MOPs, the hypervolume measure is used to assess both convergence and maximum spread of the solutions for the approximation of the Pareto front obtained with any MOEA. High values of this measure indicate that the solutions are closer to the true Pareto front and that they cover a wider extension of it.

The hypervolume contribution for a nondominanted solution in the approximated Pareto front can be estimated based on its closest neighbors in each objective dimension [23]. Figure 1 shows an example of the hypervolume contribution for a set of solutions in two dimensions. In this figure, the dominated shaded area adjacent to each solution is defined as the solution's hypervolume contribution.

A naive way to compute the hypervolume contribution for each solution, is to first compute the hypervolume of the whole Pareto set approximation, and then compute the correspondig hypervolume without the solution which the contribution is computed for. It is important to note that the extreme solutions are assigned an ∞-value contribution; otherwise, they are discarded in the selection process. It might also result obvious that the hypervolume contribution

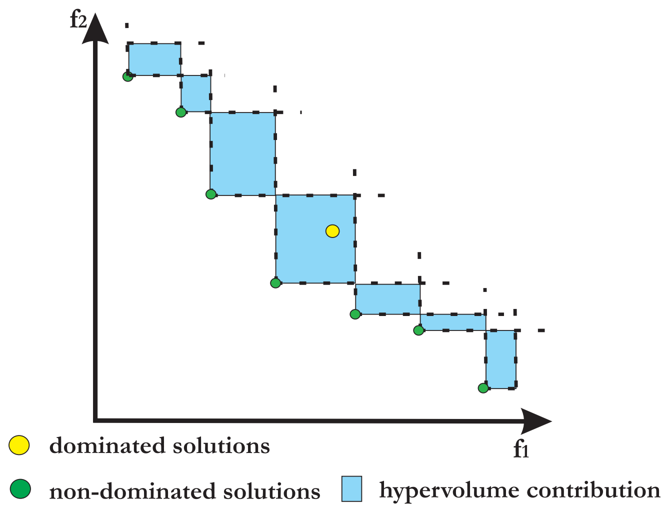

Fig. 1. *Example of Hypervolume contributions.*

calculation has a very high computational cost with an increasing number of objectives and with a high number of particles in the swarm. In this regard, different alternative approaches have been proposed to overcome this drawback. In our proposal we adopted the algorithm for estimating the hypervolume contribution using Monte Carlo simulations incorporated in HypE [24].

C. Reference point construction

The estimation of the hypervolume contribution needs a reference point z_{ref} in objective space. Obvious options for this point are the *ideal* and *nadir* vectors for a maximization and minimization MOP, respectively. Since, in general we do not know a priori these points for any given MOP, it is common to approximate them, based on the current approximation that we have of the Pareto front.

The approximation for the ideal vector is the minimum of each objective:

$$(z_{ref})_{ideal} = \left(\min_{\vec{x}\in PF_{known}} f_1\left(\vec{x}\right), \ldots, \min_{\vec{x}\in PF_{known}} f_k\left(\vec{x}\right)\right)$$

where PF_{known} contains all the known nondominated solutions to the MOP. The approximation for the nadir vector is the maximum of each objective, and is obtained by taking into account only the nondominated solutions in the current Pareto front approximation:

$$(z_{ref})_{nadir} = \left(\max_{\vec{x}\in PF_{known}} f_1\left(\vec{x}\right), \ldots, \max_{\vec{x}\in PF_{known}} f_k\left(\vec{x}\right)\right)$$

During the evolutionary process, either the ideal or the nadir reference points must be updated, whenever the Pareto front approximation changes.

D. Selection of leaders

The selection of the global leader ($gBest$) for each particle in the swarm's primary population is considered to be a crucial step in the design of any MOPSO, since it has effects both in the convergence performance as well as

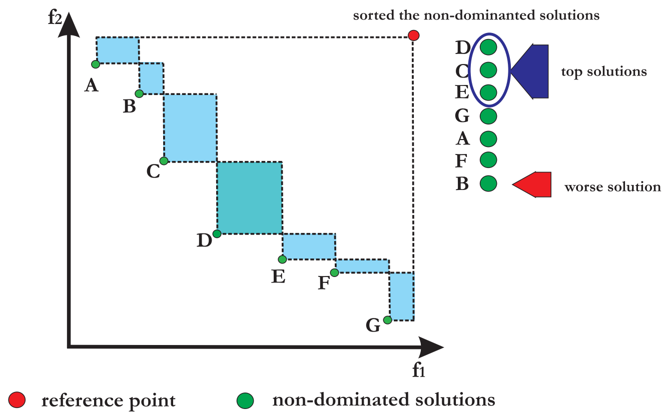

Fig. 2. *Sorted repository based on decreasing hypervolume contribution values.*

on the good spreading of nondominated solutions in the Pareto front approximation for the algorithm. As indicated before, in MOPSOhv, a bounded external archive stores the nondominated solutions found in previous iterations. We note that any of these solutions can be used as the global leader ($gBest$) for the particles in the swarm. However, in our case, we wanted to ensure that the particles in the primary population move towards the highest quality regions of the search space. Thus, the global leader in MOPSOhv is selected from the nondominated solutions having the highest hypervolume contributions values. An ilustration of this process is shown in Figure 2,

Another important aspect when designing a MOPSO is the selection of personal leaders ($pBest$) for each particle in the swarm. In MOPSOhv, the design motivation is to promote better interaction for each particle by using the bottom part of the sorted external archive repository, i.e., the selection for the personal leader aims to promote the movement of particles towards different regions with respect to the best global solution.

E. Mutation operator

In our MOPSOhv, a mutation operator is included. This operator was introduced due to its exploratory capability. In our approach, the mutation operator acts by performing more mutations at the early stages of the search process, and we rapidly reduce its use over time. This sort of mutation operator has been found to be very effective in other MOPSOs [16], [18].

IV. Performance Assessment

Our proposed MOPSOhv was validated using 12 test problems from the Zitzler-Deb Thiele (ZDT) [25] and the Deb-Thiele-Laumanns-Zitzler (DTLZ) [26] test suites. These MOPs were adopted with the settings shown in Table I.

We compared our approach with respect to NSGA-II [7] (which is a Pareto-based MOEA), SMS-EMOA [8] (which is a hypervolume-based MOEA) and MOPSOcd [18] (which is a Pareto-based MOPSO representative of the state-of-the-art in the area).

Problem	# variables	# objetives
ZDT1-3	30	2
ZDT4,6	10	2
DTLZ1	13	3
DTLZ2-6	12	3
DTLZ7	22	3

TABLE I
TEST PROBLEM SETTINGS

A. Parameters Settings

We performed 20 independent runs for each algorithm with the parameters shown in Table II. The four MOEAS adopted a population size of 100 individuals, an archive size with 100 solutions and were run during 1200 generations.

MOPSOcd	MOPSOhv	NSGA-II	SMS-EMOA
$\omega = 0.4$	$\omega = 0.4$	$P_c = 0.9$	—
$p_m = 0.5$	$p_m = 0.5$	$P_m = \frac{1}{\vec{x}}$	$P_m = \frac{1}{\vec{x}}$
$\varphi_1 = 1$	$\varphi_1 = 1$	$n_c = 20$	$n_c = 20$
$\varphi_2 = 1$	$\varphi_1 = 1$	$n_c = 20$	$n_c = 20$

TABLE II
PARAMETERS USED FOR EACH MOEA

The algorithm MOPSOcd selects the global leader from the top 10% sorted archive and replaces one of the non-dominated solutions in the bottom 10% of the archive. Our proposed MOPSOhv selects the global leader from the top 2% portion of the archive, and the local leader from the remaining 98% sorted archive. Particle replacement in the archive deletes the least hypervolume contributing particle when the maximum allowable size of the archive is exceeded. The outcome sets for the diferent MOEAs were compared using the following performance measures: spread [7] (I_s), inverse generational distance IGD [27] (IGD), and hypervolume [28] (I_{hv}) The reference points adopted for the hypervolume indicator are shown in Table III for the each of the MOPs solved.

Problem	Reference point (z_{ref})
ZDT1-3 6	$(1.1, 1.1)$
ZDT4	$(0.9, 1.1)$
DTLZ1,2,4-6	$(1.1, 1.1, 1.1)$
DTLZ3	$(5.0, 5.0, 5.0)$
DTLZ7	$(1.1, 1.1, 7.0)$

TABLE III
REFERENCE POINTS USED FOR COMPUTING THE HYPERVOLUME

V. ANALYSIS OF RESULTS

In this section, we present the comparison of the results obtained by the different MOEAs previously indicated in the selected MOPs. A summary of our results is shown in Tables IV, V and VI for the different performance indicators adopted. In these tables, the mean and standard deviation are reported for each performance measure and for each MOP.

Regarding spread (I_s), lower values indicate better performance for a given MOEA. From Table IV, it can be observed that SMS-EMOA is the best performer, obtaining the best value in eleven of the twelve MOPs adopted in our study. Our proposed approach ranks second in six MOPs, it ranks third in four MOPs, and it ranks fourth in only two MOPs. From these results and with respect to the spread of solutions, we can conclude that our proposed approach is competitive with respect to SMS-EMOA and is better than NSGA-II and MOPSOcd.

With respect to the inverted generational distance (IGD), lower values also indicate better performance for a given MOEA. From Table V, we can observe that NSGA-II ranks first in seven MOPs, SMS-EMOA in three MOPs, and MOPSOcd in two MOPs. Our proposed approach ranks third in seven MOPs. Since the IGD indicator measures convergence for a given MOEA, we can conclude that our approach is not better than the others in terms of convergence, but the values that it reached are reasonably good. So, we claim that it is competitive in terms of convergence.

Finally, with respect to the hypervolume indicator (I_{hv}), higher values indicate better performance for a given MOEA. From Table VI, it can be observed that SMS-EMOA ranks first in nine of twelve MOPs. Our proposed approach ranks second in four MOPs, third in three MOPs, and fourth in four MOPs. These results somehow reinforce our previous claims, since hypervolume assesses both convergence and spread. The resuls obtained by our approach indicate that it is competitive with respect to the other MOEAs adopted for our comparative study.

It is important to note that in our approach, the hypervolume is estimated, and not calculated in an exact manner as done by SMS-EMOA. Because of this, it was expected that our proposed approach would have a lower performance than SMS-EMOA. These hypervolume estimations, however, produce important savings in terms of computational time, as will be seen next.

We illustrate this reduction in performance by showing two graphical approximations for the Pareto fronts in solving DTLZ2, DTLZ4, and DTLZ7. In Figure 3, correspondig to DTLZ2, we can observe that SMS-EMOA attains a good convergence and distribution of solutions along the Pareto front. Our approach has a better converge as compared to MOPSOcd, and is able to cover a wider area of the Pareto front as compared to NSGA-II. In Figure 4, we also observe that SMS-EMOA is able to attain good convergence and a good distribution of solutions along the Pareto front approximation. For this MOP, our approach performed worst in terms of the spread of solutions as compared to MOPSOcd and NSGA-II. However, our proposed approach has a better convergence ability, but solutions are clustered toward the boundaries of the true Pareto front. A similar behavior is observed in other MOPs such as DTLZ7 (see Figure 5). This reduced performance in solving some MOPs deserves a further investigation, and will be part of our future work.

It is well-known that the exact computation of the hypervolume contribution used in SMS-EMOA hinders its use in many-objective optimization problems (i.e., in problems having more than 3 objectives), due to its unaffordable computational cost in those cases. In this regard, our approach,

Problem	MOPSOhv mean (σ)	MOPSOcd mean (σ)	NSGA-II mean (σ)	SMS-EMOA mean (σ)
ZDT1	0.003524 (0.000305)	0.006962 (0.000362)	0.007020 (0.000447)	**0.002320** (0.000270)
ZDT2	0.004679 (0.000217)	0.007014 (0.0005170)	0.006956 (0.000610)	**0.004063** (0.000543)
ZDT3	0.006248 (0.000328)	0.007656 (0.000425)	0.007875 (0.000565)	**0.002209** (0.000165)
ZDT4	0.004679 (0.000217)	0.007014 (0.000517)	0.008052 (0.000616)	**0.002492** (0.000240)
ZDT6	0.001975 (0.000324)	0.115691 (0.053160)	0.007055 (0.000437)	**0.000603** (0.000181)
DTLZ1	0.081790 (0.045829)	0.974734 (0.295577)	0.019360 (0.002384)	**0.006947** (0.001461)
DTLZ2	0.056173 (0.012780)	0.052077 (0.005154)	0.055460 (0.005003)	**0.042716** (0.001942)
DTLZ3	1.759328 (1.322038)	2.143857 (0.915538)	0.056032 (0.004531)	**0.043889** (0.009394)
DTLZ4	0.047999 (0.021016)	0.056542 (0.002922)	0.051862 (0.013087)	**0.042030** (0.001928)
DTLZ5	0.016552 (0.005419)	**0.008436** (0.000610)	0.009727 (0.000757)	0.008763 (0.000463)
DTLZ6	0.158278 (0.026187)	0.343938 (0.052728)	0.020970 (0.004354)	**0.013185** (0.002414)
DTLZ7	0.085100 (0.025882)	0.114527 (0.112693)	0.068548 (0.009758)	**0.057420** (0.005603)

TABLE IV

COMPARISON OF THE RESULTS OBTAINED FOR I_s BY NSGA-II, SMS-EMOA, MOPSOCD AND OUR PROPOSED MOPSOHV FOR THE ZDT AND DTLZ TEST PROBLEMS

Problem	MOPSOhv mean (σ)	MOPSOcd mean (σ)	NSGA-II mean (σ)	SMS-EMOA mean (σ)
ZDT1	0.017014 (0.000003)	0.017173 (0.000036)	0.017011 (0.000003)	**0.017009** (0.000001)
ZDT2	0.027391 (0.000001)	0.027497 (0.000015)	0.027387 (0.000001)	**0.027386** (0.000000)
ZDT3	0.011198 (0.000012)	0.011492 (0.000064)	**0.011166** (0.000016)	0.019892 (0.000001)
ZDT4	0.027391 (0.000001)	0.027497 (0.000015)	0.017011 (0.000002)	**0.017011** (0.000004)
ZDT6	0.027434 (0.000046)	**0.000281** (0.000000)	0.026356 (0.003115)	0.027395 (0.000002)
DTLZ1	0.013359 (0.011381)	0.269641 (0.080597)	**0.001261** (0.001148)	0.006088 (0.001259)
DTLZ2	0.000806 (0.000611)	0.000383 (0.000022)	**0.000380** (0.000018)	0.005000 (0.000000)
DTLZ3	0.594644 (0.302257)	0.630432 (0.323953)	**0.001235** (0.000083)	0.016425 (0.003406)
DTLZ4	0.006694 (0.002366)	**0.001208** (0.000037)	0.001875 (0.003071)	0.015606 (0.000000)
DTLZ5	0.000439 (0.000390)	**0.000054** (0.000002)	0.000069 (0.000005)	0.009901 (0.000000)
DTLZ6	0.009931 (0.002293)	0.036889 (0.012547)	**0.000648** (0.000236)	0.010786 (0.000231)
DTLZ7	0.002800 (0.001392)	0.001558 (0.000205)	**0.001123** (0.000957)	0.015064 (0.002691)

TABLE V

COMPARISON OF THE RESULTS OBTAINED FOR I_{IGD} INDICATOR BY NSGA-II, SMS-EMOA, MOPSOCD AND OUR PROPOSED MOPSOHV FOR THE ZDT AND DTLZ TEST PROBLEMS

which uses a Monte Carlo estimation of the hypervolume contribution could be a viable choice for dealing with such many-objective problems. In order to assess this, we present next a small scalability test.

A. Many-objetive Optimization: A Case Study

In order to evaluate the ability of our proposed approach to deal with many-objective optimization problems, we scaled

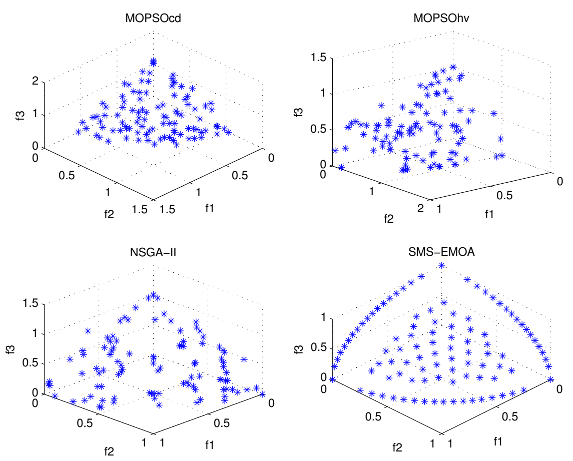

Fig. 3. *Graphical results for DTLZ2.*

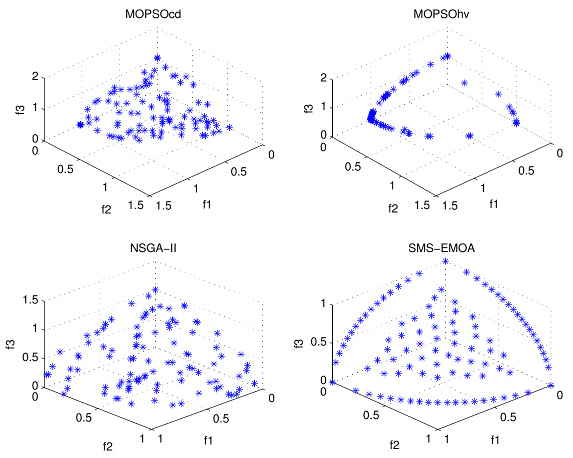

Fig. 4. *Graphical results for DTLZ4.*

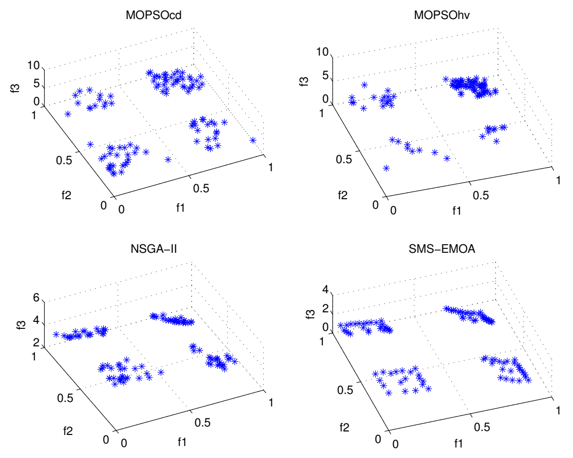

Fig. 5. *Graphical results for DTLZ7.*

Problem	MOPSOhv	MOPSOcd	NSGA-II	SMS-EMOA
	mean	mean	mean	mean
	(σ)	(σ)	(σ)	(σ)
ZDT1	0.871849	0.864645	0.870461	**0.872129**
	(0.000095)	(0.001562)	(0.000206)	(0.000006)
ZDT2	0.538568	0.532098	0.537424	**0.538872**
	(0.000058)	(0.000882)	(0.000148)	(0.000015)
ZDT3	0.952907	0.941991	**0.953912**	0.522464
	(0.000452)	(0.002510)	(0.000145)	(0.000001)
ZDT4	0.328971	0.324090	0.653614	**0.654940**
	(0.000052)	(0.000668)	(0.000237)	(0.000165)
ZDT6	0.512122	0.383107	**0.519667**	0.503956
	(0.002083)	(0.076527)	(0.117011)	(0.000093)
DTLZ1	0.838140	N/A	1.270538	**1.295857**
	(0.351522)	(N/A)	(0.016839)	(0.039586)
DTLZ2	0.626972	0.669615	0.697212	**0.758091**
	(0.046583)	(0.019100)	(0.007736)	(0.000047)
DTLZ3	N/A	N/A	123.673815	**124.221079**
	(N/A)	(N/A)	(0.295556)	(0.878873)
DTLZ4	0.548703	0.688920	0.674632	**0.758069**
	(0.117426)	(0.010033)	(0.127135)	(0.000050)
DTLZ5	0.428426	0.438852	0.437990	**0.439374**
	(0.010028)	(0.000056)	(0.000223)	(0.000011)
DTLZ6	0.000391	N/A	**0.361529**	0.340557
	(0.000277)	(N/A)	(0.028920)	(0.024323)
DTLZ7	2.608814	2.648845	2.974631	**5.456793**
	(0.123164)	(0.072522)	(0.091615)	(0.168653)

TABLE VI

COMPARISON OF THE RESULTS OBTAINED FOR I_{hv} BY NSGA-II, SMS-EMOA, MOPSOCD AND OUR PROPOSED MOPSOHV FOR THE ZDT AND DTLZ TEST PROBLEMS

Objectives	MOPSOhv	MOPSOcd	NSGA-II	SMS-EMOA
	mean	mean	mean	mean
	(σ)	(σ)	(σ)	(σ)
	avg time (s)	avg time (s)	avg time (s)	avg time (s)
2	0.420962	0.420372	0.419588	**0.421023**
	(0.000031)	(0.000037)	(0.000323)	(0.00003)
	150.6620	87.2790	1.4545	75.2620
3	0.644946	0.667862	0.697689	**0.758071**
	(0.030303)	(0.024190)	(0.009213)	(0.000056)
	241.3620	106.2830	1.6804	1066.4680
4	0.649871	0.000000	0.834518	**1.044734**
	(0.046065)	(0.000000)	(0.019843)	(0.000030)
	239.3730	145.4090	1.8407	11149.0560
5	0.686608	0.000000	**0.806271**	N/A
	(0.091850)	(0.000000)	(0.075088)	(N/A)
	280.1500	170.2110	2.0637	N/A
6	**0.768027**	0.000000	0.195606	N/A
	(0.106912)	(0.000000)	(0.133364)	(N/A)
	327.2390	247.5470	2.2874	N/A
7	**0.897171**	0.000000	0.146724	N/A
	(0.059289)	(0.000000)	(0.119704)	(N/A)
	529.7890	288.0450	2.5823	N/A
8	**0.923948**	0.000000	0.185958	N/A
	(0.098939)	(0.000000)	(0.088435)	(N/A)
	582.9530	325.6510	2.8558	N/A
9	**0.894945**	0.000000	0.212041	N/A
	(0.098291)	(0.000000)	(0.124973)	(N/A)
	653.3080	395.4500	3.0805	N/A
10	**1.080952**	0.000000	0.224003	N/A
	(0.085351)	(0.000000)	(0.127577)	(N/A)
	382.7230	335.1710	3.2584	N/A

TABLE VII

COMPARISON OF THE RESULTS OBTAINED FOR I_{hv} BY NSGA-II, SMS-EMOA, MOPSOCD AND OUR PROPOSED MOPSOHV FOR DTLZ2 USING FROM 2 TO 10 OBJECTIVES. THE AVERAGE EXECUTION TIMES ARE ALSO REPORTED.

the DTLZ2 test problem from 2 to 10 objectives. As before, we compared the performance of our proposed approach with respect to NSGA-II, SMS-EMOA and MOPSOcd. The parameters used in this case are the same values reported in Section IV-A.

The outcome sets for the algorithms were compared using the hypervolume (I_{hv}) indicator, using $1.\vec{1}$ as the reference vector (z_{ref}). Additionally, the running time for each the different MOEAs compared is reported. We can see in Table VII and in Figure 6 how, as expected for Pareto-based MOEAs, NSGA-II and MOPSOcd have both a quick performance degradation (with respect to the hypervolume) as we increase the number of objectives. This does not occur with SMS-EMOA nor with our proposed MOPSOhv. SMS-EMOA attains better solutions than our proposal, but at a much higher computational cost, see Figure 7. In these experiments, the use of SMS-EMOA was limited to four objectives, due to the exponential increase in its computational cost. From these results, we can see how our proposed MOPSOhv is a viable alternative for solving many-objective optimization problems, since its computational cost remains affordable even when dealing with ten objectives. In fact, our approach was able to improve the quality of the solutions obtained as the number of objectives was increased.

VI. CONCLUSIONS AND FUTURE WORK

Since the leaders selection strategy strongly influence the performance of multi-objective particle swarm optimizers, we have proposed here a mechanism based on the hypervolume contribution of the archived particles for this sake. Also, and due to the high computational cost involved in evaluating exact hypervolume contributions we approximated them

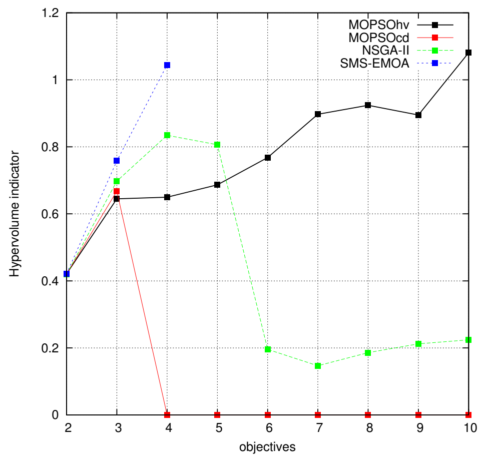

Fig. 6. *Indicator values for DTLZ2.*

using Monte Carlo simulations. From the experiments performed, it can be seen that our proposed approach produces competitive results in low dimensionality and it outperforms state-of-the-art MOEAs in high dimensionality while keeping a low computational cost. Thus, we suggest its use mainly for solving many-objective optimization problems.

As part of our future work, we plan to test different schemes and topologies for the global and local leader selection schemes of our approach. We would also like to use different density estimators in the deletion process adopted in the external archive update mechanism. Finally, we are also interested in analyzing the reasons for which our approach

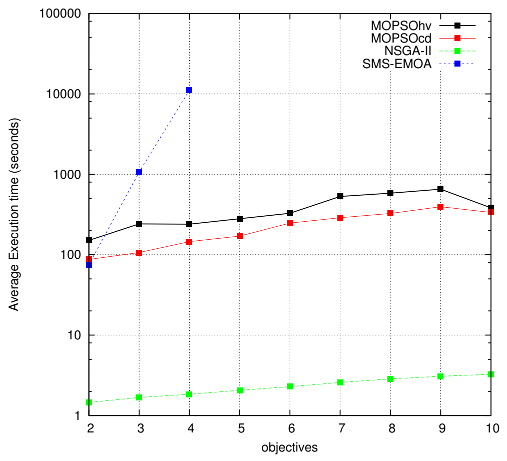

Fig. 7. *Average execution time for DTLZ2 with varying objectives.*

has problems to converge in some test problems.

ACKNOWLEDGEMENTS

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. 221551. The third author acknowledges support from IPN-ESIME through the COTEBAL, EDI and SIBE programs.

REFERENCES

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

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

[3] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

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

[5] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

[6] Eckart Zitzler, Marco Laumanns, and Lothar 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.

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

[8] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

[9] Russell C. Eberhart, Yuhui Shi, and James Kennedy. *Swarm Intelligence*. Morgan Kaufmann, 1st edition, 2001.

[10] K.E. Parsopoulos and M.N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002)*, pages 603–607, Madrid, Spain, 2002. ACM Press.

[11] Xiaohui Hu and Russell Eberhart. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1677–1681, Piscataway, New Jersey, May 2002. IEEE Service Center.

[12] Xiaohui Hui, Russell C. Eberhart, and Yuhui Shi. Particle Swarm with Extended Memory for Multiobjective Optimization. In *2003 IEEE Swarm Intelligence Symposium Proceedings*, pages 193–197, Indianapolis, Indiana, USA, April 2003. IEEE Service Center.

[13] Sanaz Mostaghim and Jürgen Teich. Covering Pareto-optimal Fronts by Subswarms in Multi-objective Particle Swarm Optimization. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 2, pages 1404–1411, Portland, Oregon, USA, June 2004. IEEE Service Center.

[14] Saúl Zapotecas Martínez and Carlos A. Coello Coello. A Multi-objective Particle Swarm Optimizer Based on Decomposition. In *2011 Genetic and Evolutionary Computation Conference (GECCO'2011)*, pages 69–76, Dublin, Ireland, July 12-16 2011. ACM Press.

[15] Sanaz Mostaghim, Jürgen Branke, and Hartmut Schmeck. Multi-Objective Particle Swarm Optimization on Computer Grids. In Dirk Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 869–875, London, UK, July 2007. ACM Press.

[16] Carlos A. Coello Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga. Handling Multiple Objectives With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[17] Maximino Salazar-Lechuga and Jonathan E. Rowe. Particle Swarm Optimization and Fitness Sharing to solve Multi-Objective Optimization Problems. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 2, pages 1204–1211, Edinburgh, Scotland, September 2005. IEEE Service Center.

[18] Carlo R. Raquel and Prospero C. Naval, Jr. An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In Hans-Georg Beyer et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 257–264, New York, USA, June 2005. ACM Press.

[19] Jambhlekar Pushkar Arun, Manoj Mishra, and Sheshasayee V. Subramaniam. Parallel implementation of MOPSO on GPU using OpenCL and CUDA. In *2011 18th International Conference on High Performance Computing (HiPC)*, Bangalore, India, December 18-21 2011. IEEE.

[20] Sanaz Mostaghim, Andrew Lewis Jürgen Branke, and Hartmut Schmeck. Parallel Multi-Objective Optimization using Master-Slave Model on Heterogeneous Resources. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 1981–1987, Hong Kong, June 2008. IEEE Service Center.

[21] You Zhou and Ying Tan. GPU-Based Parallel Multi-objective Particle Swarm Optimization. *International Journal of Artificial Intelligence*, 7(A11):125–141, October 2011.

[22] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, March 2011.

[23] Karl Bringmann and Tobias Friedrich. Don't be Greedy when Calculating Hypervolume Contributions. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 103–112, Orlando, Florida, USA, January 2009. ACM.

[24] Johannes Bader and Eckart Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring, 2011.

[25] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[26] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[27] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[28] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.