Geometric Particle Swarm Optimization for Multi-objective Optimization Using Decomposition

Saúl Zapotecas-Martínez
Shinshu University
Faculty of Engineering
4-17-1 Wakasato, Nagano,
380-8553, Japan
saul.zapotecas@gmail.com

Alberto Moraglio
University of Exeter
Department of Computer Science
Exeter EX4 4QF, UK
a.moraglio@exeter.ac.uk

Hernán E. Aguirre
Shinshu University
Faculty of Engineering
4-17-1 Wakasato, Nagano,
380-8553, Japan
ahernan@shinshu-u.ac.jp

Kiyoshi Tanaka
Shinshu University
Faculty of Engineering
4-17-1 Wakasato, Nagano,
380-8553, Japan
ktanaka@shinshu-u.ac.jp

ABSTRACT

Multi-objective evolutionary algorithms (MOEAs) based on decomposition are aggregation-based algorithms which transform a multi-objective optimization problem (MOP) into several single-objective subproblems. Being effective, efficient, and easy to implement, Particle Swarm Optimization (PSO) has become one of the most popular single-objective optimizers for continuous problems, and recently it has been successfully extended to the multi-objective domain. However, no investigation on the application of PSO within a multi-objective decomposition framework exists in the context of combinatorial optimization. This is precisely the focus of the paper. More specifically, we study the incorporation of Geometric Particle Swarm Optimization (GPSO), a discrete generalization of PSO that has proven successful on a number of single-objective combinatorial problems, into a decomposition approach. We conduct experiments on *many*-objective 1/0 knapsack problems i.e. problems with more than three objectives functions, substantially harder than multi-objective problems with fewer objectives. The results indicate that the proposed multi-objective GPSO based on decomposition is able to outperform two version of the well-know MOEA based on decomposition (MOEA/D) and the most recent version of the non-dominated sorting genetic algorithm (NSGA-III), which are state-of-the-art multi-objective evolutionary approaches based on decomposition.

Keywords

Multi-objective Combinatorial Optimization; Decomposition-based MOEAs; Particle Swarm Optimization.

GECCO '16, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908812.2908880

1. INTRODUCTION

Particle swarm optimization (PSO) [13] is a bio-inspired metaheuristic for continuous optimization problems, which has been applied very successfully to many engineering and scientific problems. This has motivated researchers to extend it to multi-objective optimization problems (MOPs). In the last decade, several multi-objective particle swarm optimizers (MOPSOs) have been developed (see [26] for a good survey on this topic). Most of these approaches use a set of best non-dominated solutions to steer the search, rather than a single global optimum as in the traditional PSO, along with an additional mechanism to maintain population diversity during the search. These approaches became very popular in the early days of MOPSOs.

The dominance resistance phenomenon [24], which is commonly observed in problems with more than 3 objectives (known as *many*-objective problems), hinders performances of MOPSOs using Pareto dominance relation by making selection ineffective. Researchers have then focused on developing alternative selection mechanisms to deal with this drawback, such as for example methods based on indicators [6, 15]. Recently, researchers working on MOEAs have adopted the idea of decomposing a MOP into several optimization subproblems. This approach has become one of the most useful strategies to deal with MOPs, especially for many-objective problems, see for instance [31, 9]. In this approach, a set of approximate solutions to the Pareto optimal front is achieved by minimizing each single-objective subproblem, rather than using Pareto optimality or alternative selection mechanisms. This trend has also led to a new generation of multi-objective particle swarm optimizers based on decomposition studied by several authors, see for example [23, 21, 33]. To date, these approaches have focused on continuous and unconstrained problems, leaving the discrete case as an open field to be explored. This is precisely the focus of the work reported herein.

Several extensions of PSO to discrete spaces have been

introduced to date, the majority of these operating on binary strings, see e.g. [14, 1, 18, 22]. Extensions of PSO to more complex combinatorial search spaces, such as permutations or TSP tours, are rarer but do exist, see e.g. [7]. The difficulty here lays in defining meaningful notions of motion, direction, and velocity in such spaces. Geometric particle swarm optimization (GPSO) [19] is a generalization of traditional particle swarm optimization to general metric spaces. These notions and the PSO algorithm dynamics are defined in this general abstract setting. Specific instantiations of GPSO can then be formally derived by using specific distances and associated solution representations in the general definition of GPSO. This approach has the advantage that PSO for specific representations can be derived in a principled way, rather than reinvented and adapted ad-hoc to each new representation. Representation-specific GPSO have been derived for binary strings [19], permutations and applied to solving Sudoku [20], and tree structures and used as an alternative search strategy for genetic programming [29]. The binary GPSO has been successfully adopted in several applications, e.g. [2, 3, 11, 28, 25].

In this paper, we introduce a new multi-objective particle swarm optimizer for combinatorial problems. The proposed approach extends the binary GPSO to work with MOPs adopting the decomposition approach. The study presented here indicates that the proposed approach is efficient and produces a good approximation to the Pareto front on multi-objective knapsack problems with a number of objectives ranging from two to ten. It is also found to be significantly better than the well-known multi-objective evolutionary algorithm based on decomposition (MOEA/D) [34] and than the most recent version of the non-dominated sorting genetic algorithm (NSGA-III) [9].

2. BASIC CONCEPTS

2.1 Preliminaries of Multi-objective Optimization

Assuming maximization, a general *multi-objective optimization problem* (MOP) can be stated as:

$$\begin{array}{rll} \text{maximize:} & \mathbf{F}(\mathbf{x}) & \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0, & i = 1, \ldots, p \\ & h_j(\mathbf{x}) = 0, & j = 1, \ldots, q \\ & \mathbf{x} \in X & \end{array} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^\intercal$ is an n dimensional vector of decision variables. The vector $\mathbf{F} = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))^\intercal$ consists of M objective functions f_j's to be maximized. $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$ represent the p inequality constraints and the q equality constraints, respectively. The set of solutions that satisfy the constraints of problem (1) defines the feasible region $\Omega \subset X$. When problem (1) is continuous $X = \mathbb{R}^n$. In the case of pseudo-boolean combinatorial problems the search space is $X = \{0, 1\}^n$.

The following definitions introduce the concept of optimality of interest in this paper (see [17]).

DEFINITION 1. *Let* $\mathbf{x}, \mathbf{y} \in \Omega$, *we say that* $\mathbf{x}$ dominates $\mathbf{y}$ *(denoted by* $\mathbf{x} \succ \mathbf{y}$*) if and only if:* 1) $f_i(\mathbf{x}) \geq f_i(\mathbf{y})$ *for all* $i \in \{1, \ldots, M\}$ *and* 2) $f_j(\mathbf{x}) > f_j(\mathbf{y})$ *for at least one* $j \in \{1, \ldots, M\}$.

DEFINITION 2. *Let* $\mathbf{x}^\star \in \Omega$, *we say that* $\mathbf{x}^\star$ *is a* Pareto optimal *solution, if there is no other solution* $\mathbf{y} \in \Omega$ *such that* $\mathbf{y} \succ \mathbf{x}^\star$.

DEFINITION 3. *The* Pareto optimal set PS *is defined by:* $PS = \{\mathbf{x} \in \Omega | \mathbf{x}$ is a Pareto optimal solution$\}$ *and its image* $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in PS\})$ *is called* Pareto front PF.

In multi-objective optimization problems, we are typically interested in finding a finite number of elements from the Pareto set, while maintaining a proper representation of the Pareto front.

2.2 Decomposition of a Multi-objective Optimization Problem

It is well-known [17] that a Pareto optimal solution to the problem (1) is an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objective functions f_i's. Many scalar approaches have been proposed to aggregate the objectives of a MOP. Among them, the Tchebycheff approach [5] is one of the most used methods, and it is the one adopted in this study. Other scalarization approaches could also be easily used, see e.g., [10, 17].

Tchebycheff approach. This approach transforms the vector of function values $\mathbf{F}$ into a scalar maximization problem which is of the form:

$$\begin{array}{rll} \text{minimize} & g^{tch}(\mathbf{x}|\lambda, \mathbf{z}) = \max\limits_{1 \leq j \leq M} \{\lambda_j |z_j - f_j(\mathbf{x})|\} & \\ s.t. & \mathbf{x} \in \Omega & \end{array} \tag{2}$$

where Ω is the feasible region, $\mathbf{z} = (z_1, \ldots, z_k)^\intercal$ is the reference point such that $z_j = \max\{f_j(\mathbf{x}) | \mathbf{x} \in \Omega\}$ for each $i = 1, \ldots, M$, and $\lambda = (\lambda_1, \ldots, \lambda_M)^\intercal$ is a weight vector, i.e., $\lambda_j \geq 0$ for all $j = 1, \ldots, M$ and $\sum_{j=1}^{M} \lambda_j = 1$.

For each Pareto optimal point $\mathbf{x}^\star$, there exists a weight vector λ such that $\mathbf{x}^\star$ is the optimum solution of equation (2) and each optimal solution of equation (2) is a Pareto optimal solution of equation (1). An appropriate representation of the Pareto front could be reached by solving different scalarizing problems. Such problems can be defined by a set of well-distributed weight vectors, which establish the search direction in the optimization process.

Therefore, a proper approximation to the Pareto front can be reached by minimizing a set of scalarizing functions defined by a well-distributed set of weights vectors. This is the main idea behind mathematical programming methods for multi-objective problems and current multi-objective evolutionary approaches based on decomposition, e.g. [34, 23, 16].

2.3 Geometric Particle Swarm Optimization

The generalization of the standard PSO algorithm for continuous spaces to general metric spaces is based on the following idea (see [19] for more details and the mathematical derivation). The only elements of the standard PSO algorithm that depend on the underlying representation are the velocity update and position update equations which require velocities and positions to be real vectors. The velocity update equation can be factored out and equivalently restated in terms of current and past positions of each particle. In absence of inertia, the new position of a particle can be written as a convex combination of its current position, its

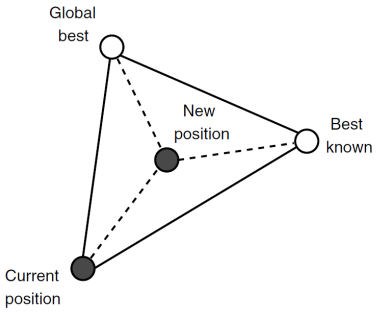

Figure 1: The convex combination operator moves the particle based on its current position, position of its personal best and position of the global best.

personal best position, and the position of the global best (see figure 1).

As the notion of convex combination is well-defined in general metric spaces, the PSO algorithm can then be readily generalized to metric spaces. The generic Geometric PSO algorithm is illustrated in Algorithm 1. This differs from the standard PSO in that: (i) there is no explicit velocity update equation (but particles have velocities as they move); (ii) the equation of the position update is a (randomized) convex combination as outlined above, with weights ω, ϕ_1, and ϕ_2, which are are non-negative and add up to one; (iii) the new position undergoes to mutation to partly compensate for the lack of inertia.

The specific PSO for the space of binary strings [1] endowed with Hamming distance can be obtained by formally deriving an explicit definition of randomized convex combination for this space in terms of manipulation of the underlying representation. This operator then specifies operationally how to obtain an 'offspring' binary string which corresponds to the convex combination of 'parent' binary strings. In [19], this operator was shown to be a straightforward generalization of mask-based crossover for two parents to a three-parental recombination, in which the probability of inheriting a bit at each position from a parent string is given by its weight in the convex combination. The mutation employed in binary GPSO is standard bit-wise mutation.

Algorithm 1: Geometric PSO algorithm

```
for each particle i do
    initialize position x_i at random in the search space
while stop criteria not met do
    for each particle i do
        set personal best x̂_i as best position found so far by the particle
        set global best ĝ as best position found so far by the whole swarm
    for each particle i do
        update position using a randomized convex combination:
            x_i = CX((x_i, ω), (ĝ, φ_1), (x̂_i, φ_2))        (3)
        mutate x_i
```

3. MULTI-OBJECTIVE GPSO BASED ON DECOMPOSITION

Our proposed multi-objective GPSO based on decomposition (MO-GPSO/D) decomposes a MOP into several single-objective subproblems by using an aggregation function and a weight vector, as in MOEA/D [34]. However, the proposed approach does not follow the principles of MOEA/D. The main differences between MOEA/D and MO-GPSO/D are the recombination and replacement strategies.

Recombination strategy. MOEA/D defines a neighborhood in order to select random solutions to be recombined. For binary combinatorial problems, MOEA/D adopts traditional operators taken from genetic algorithms (specifically, one-point crossover, and bit-wise mutation), in order to generate candidate solutions, see [34]. This strategy works well in continuous problems. However, in the case of combinatorial problems (where the fitness landscape is unknown) the regulatory property of continuous MOPs (the idea behind of MOEA/D [34]) does not claim that an optimal solution of a subproblem with weight λ^1, is close to other optimal solution of other subproblem with weight λ^2, for $||\lambda^1 - \lambda^2|| < \epsilon$ (for an ϵ small enough), i.e. neighboring subproblems. Therefore we hypothesize that MO-GPSO/D could work better than MOEA/D if the recombination of solutions is not restricted into a single neighborhood as in MOEA/D. MO-GPSO/D uses the geometric version of PSO to create new solutions by using the *personal best* (the best position of the particle to the ith subproblem) and the *global best* solution (solution found from all the swarm which achieves the best value for the ith subproblem along the search).

Replacement strategy. MOEA/D replaces *all* solutions in the neighborhood which are improved by the new candidate solution. This mechanism works well for MOPs having relatively easy Pareto sets [16]. However, for more complex problems (see for example [35]), this strategy becomes inefficient and sometimes impractical. In fact, this strategy can misplace diversity in the population specially in multimodal problems or problems with rugged landscapes. This drawback was treated in [16] where a dynamic neighborhood selection and a maximum number replacements were implemented, arising in a new version of MOEA/D namely MOEA/D-DE. In MO-GPSO/D, the replacement of global best solutions is carried out in different way. We do not replace all the improved solutions with the new one. Instead of this, the new solution is in competition with the current best solutions and from them, the new set of global best solutions is defined. Therefore, we speculate that this mechanism could maintain more diversity in the population, which is especially important in multi-objective optimization, while at the same time, GPSO steers the search towards promising regions employing the best solutions found along the search.

The pseudocode of the proposed MO-GPSO/D is presented in Algorithm 2. To follow a decomposition of a MOP, a well-distributed set of weighted vectors $\Lambda = \{\lambda^1, \ldots, \lambda^N\}$ has to be defined before running the algorithm. The com-

[1] A Python implementation of this algorithm is available at https://github.com/amoraglio/.

plete algorithm works as follows.

At the beginning of the algorithm, the set of the positions of the N particles $P = \{\mathbf{p}^1, \ldots, \mathbf{p}^N\}$ is randomly initialized. In MO-GPSO/D, the ith particle is set to optimize one of the subproblems defined by the weighted vector λ^i. To this end, in the main cycle, each particle 'flies' towards a better position for its single particular subproblem, i.e., the one with objective function $g^{\text{tch}}(\mathbf{p}^i|\lambda^i, \mathbf{z}^\star)$ for the ith particle.

The best personal position is initialized with the initial position of the particle, i.e., $\hat{\mathbf{p}}^i = \mathbf{p}^i$. On the other hand, the set of *global best* positions G_{best} is stated by the initial positions P, i.e. $G_{best} = P$.

The position of each particle is updated by using the recorded personal best and the global best of each particle. Then the bitwise mutation is employed as turbulence operator on each particle. Once a new position is computed, the reference point needs to be updated (line 10 in Algorithm 2). Then, the personal best $\hat{\mathbf{p}}^i$ is updated if the new position improves the previous position (line 12 in Algorithm 2).

Throughout the search, the set of global bests (denoted by $G_{best} = \{\hat{\mathbf{g}}^1, \ldots, \hat{\mathbf{g}}^N\}$) shall contain the solutions that optimize each separate subproblem. This set of solutions is then updated when a new candidate solution is generated (line 17 in Algorithm 2). Thus, the notion of elitism used in evolutionary multi-objective optimization is implicitly employed in our proposed approach.

The proposed approach tries to optimize a set of subproblems whose final solutions should be very close to the Pareto optimal set. We expect that all solutions in G_{best} are equally good i.e., that all the subproblems will be approximately satisfactorily solved, as the same search procedure and search effort is applied to all of them. Therefore, at the end of the search this set of solutions is considered as the final approximation to the Pareto optimal set.

Algorithm 2: General Framework of MO-GPSO/D

Input:
N: the number of subproblems to be decomposed;
Λ: a well-distributed set of weight vectors $\{\lambda^1, \ldots, \lambda^N\}$;
Output:
P: the final approximation to the Pareto set.

1 $\mathbf{z} = (-\infty, \ldots, -\infty)^\intercal$;
2 Generate a random set of solutions $P = \{\mathbf{p}^1, \ldots, \mathbf{p}^N\}$ in Ω;
3 $G_{best} = P$;
4 **for** $i = 1, \ldots, N$ **do**
5 $\quad \hat{\mathbf{p}}^i = \mathbf{p}^i$;
6 $\quad z_j = \max(z_j, f_j(\mathbf{p}^i))$; // $j \in \{1, \ldots, M\}$
7 **while** *stopping criterion is not satisfied* **do**
8 $\quad$ **for** $\mathbf{p}^i \in P$ **do**
9 $\quad\quad$ Update Position: Update position using a randomized convex combination:
$$\mathbf{p}^i = \text{CX}((\mathbf{p}^i, \omega), (\hat{\mathbf{g}}^i, \phi_1), (\hat{\mathbf{p}}^i, \phi_2))$$
10 $\quad\quad$ Turbulence: Apply turbulence operator to the new particle:
$$\mathbf{p}^i = \texttt{mutation}(\mathbf{p}^i)$$
11 $\quad\quad$ Update z: Update the reference point $\mathbf{z}$:
$z_j = \max(z_j, f_j(\mathbf{p}^i))$; // $j \in \{1, \ldots, M\}$
12 $\quad\quad$ Update Personal Best:
13 $\quad\quad$ If $g^{\text{tch}}(\mathbf{p}^i|\lambda^i, \mathbf{z}) \geq g^{\text{tch}}(\hat{\mathbf{p}}^i|\lambda^i, \mathbf{z})$ then $\hat{\mathbf{p}}^i = \mathbf{p}^i$;
14 $\quad\quad$ Update Global Bests:
15 $\quad\quad$ $Q = G_{best} \cup \{\mathbf{p}^i\}$;
16 $\quad\quad$ **for** $j \in \{1, \ldots, N\}$ **do**
17 $\quad\quad\quad$ $\hat{\mathbf{q}} = \arg\max_{\mathbf{q} \in Q} g^{\text{tch}}(\mathbf{q}|\lambda^j, \mathbf{z})$;
18 $\quad\quad\quad$ $\hat{\mathbf{g}}^j = \hat{\mathbf{q}}$;
19 $\quad\quad\quad$ $Q = Q \setminus \{\hat{\mathbf{q}}\}$;
20 **return** $G_{best} = \{\hat{\mathbf{g}}^1, \ldots, \hat{\mathbf{g}}^N\}$;

4. EXPERIMENTAL DESIGN

4.1 Multi-Objective 0/1 Knapsack Problem

In order to test the performance of the proposed MO-GPSO/D, the knapsack problem, one of the most studied NP-hard problems from combinatorial optimization, is adopted in a multi-objective optimization context.

Given a collection of n items and a set of M knapsacks, the multi-objective 0/1 knapsack problem (MO-KNP) seeks a subset of items subject to capacity constraints based on a *weight function* vector $w : [0,1]^n \to \mathbb{N}^M$, while maximizing a *profit function* vector $p : [0,1]^n \to \mathbb{N}^M$. Formally it can be stated as:

$$\begin{array}{rll} \text{maximize:} & f_j(\mathbf{x}) = \sum_{i=1}^n p_{ji} \cdot x_i & j \in \{1, \ldots, M\} \\ \text{s.t.} & \sum_{i=1}^n w_{ji} \cdot x_i \leqslant c_j & j \in \{1, \ldots, M\} \\ & x_i \in \{0, 1\} & i \in \{1, \ldots, n\} \end{array} \quad (4)$$

where $p_{ji} \in \mathbb{N}$ is the profit of item i on knapsack j, $w_{ji} \in \mathbb{N}$ is the weight of item i on knapsack j, and $c_j \in \mathbb{N}$ is the capacity of knapsack j.

We consider the conventional instances proposed in [37], with random uncorrelated profit and weight integer values taken uniformly from $[10, 100]$. The capacity is set to half of the total weight of a knapsack for each objective function, i.e. $c_j = \frac{1}{2}\sum_{i=1}^n w_{ji}$ for $j = 1, \ldots, M$. As a result, about 50% of the items are expected to be in the Pareto optimal front.

Random problem instances of 128 items are investigated for each objective space dimension. We consider instances with 2, 3, 5, 8, and 10 objectives[2].

In order to satisfy the constraints of the problem, we adopt a standard decoding procedure which guarantees the feasibility of solutions as proposed in [37]. This procedure removes items sorted in increasing order of the maximum profit/weigh ratio over all knapsacks one at a time, until all constraints are satisfied.

4.2 Experimental Setup

We compare experimentally MO-GPSO/D with three state-of-the-art MOEAs based on decomposition: the original MOEA/D [34], its new variant MOEA/D-DE [16] adapted to binary spaces (as explained below) and NSGA-III [9].

The method MOEA/D-DE extends MOEA/D as follows: it uses dynamic selection of the neighborhood with a given probability δ, and a fixed maximum number of replacements n_r in the neighborhood. Furthermore, MOEA/D-DE uses Differential Evolution (DE) operator as reproduction mechanism, which is however defined for continuous spaces. To adapt this algorithm to binary string spaces, the DE operator is replaced with standard crossover and mutation operators for binary strings, as in the original version of

[2]The set of instances adopted in our comparative study are available at http://computacion.cs.cinvestav.mx/~zapoteca/MO-KNP/

Table 1: Configuration for the two-layered simplex-lattice design

M (objectives)	Layer	Layer configuration	Number of weights
2	1	$H = 99$	100
3	1	$H = 19$	210
5	1	$H = 6$	210
8	2	$H_1 = 3, H_2 = 2$	156
10	2	$H_1 = 3, H_2 = 2$	275

MOEA/D, see [34]. We adopted this variant in order to test the performances of MOEA/D with dynamic selection and the limit on the number of replacements. In the rest of the paper, we refer to this variant as MOEA/D*.

For fairness, the set of weight vectors for all the algorithms in the comparison was the same, and it was generated using the Simplex-lattice design [27], as follows. The settings of N (number of weights and population size) and $\Lambda = \{\lambda^1, \ldots, \lambda^N\}$ are controlled by a parameter H. More precisely, $\lambda^1, \ldots, \lambda^N$ are weight vectors whose component scalar weights λ^i_j ($i = 1, \ldots, N$ and $j = 1, \ldots, M$) take values in $\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}$. Therefore, the number of all possible choice of vectors in Λ is given by $N = C^{M-1}_{H+M-1}$, where M is the number of objective functions. Since, this number increases binomially with the number of objectives, this methodology becomes quickly impractical when we have more than a handful of objectives. A strategy to deal with high dimensional spaces is proposed in [9], known as the two-layered simplex-lattice design. This strategy uses the simplex-lattice design to generate an outside layer and an inside layer in the weights set. Fig. 2 illustrates the two-layered simplex-lattice design in $\mathbb{R}^3$ when using $H_1 = 2$ for the outside layer and $H_2 = 1$ for the inside layer. In this study, we compare the decomposition-based approaches by using the weights given by the two-layered simplex-lattice design for problems with more than 5 objectives, otherwise a single layer is employed. The complete configuration of H values for different dimensions of the two-layered simplex-lattice design is shown in Table 1.

In our comparison, MOEA/D, MOEA/D*, and NSGA-III use the same reproduction operators, one-point crossover and bit-wise mutation, as in the original version of MOEA/D [34]. MO-GPSO/D uses also bit-wise mutation as turbulence operator.

Table 2 presents the parameter settings used in our experimental study. The parameters for the adopted algorithms

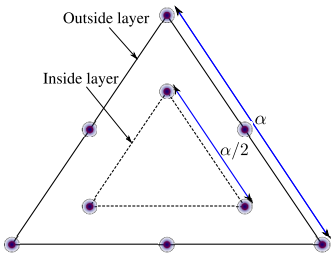

Figure 2: Illustration of the two-layered simplex-lattice design. The outside layer is stated by $H_1 = 2$ (generating six weights vectors), while the inside layer is set by $H_2 = 1$ (generating three weights vectors)

Table 2: Parameters for MO-GPSO/D, MOEA/D, MOEA/D*, and NSGA-III

Parameter	MO-GPSO/D	MOEA/D	MOEA/D*	NSGA-III
T	—	20	20	—
δ	—	—	0.9	—
n_r	—	—-	2	—
P_c	—	1	1	1
P_m	$1/n$	$1/n$	$1/n$	$1/n$
ω	1/3	—-	–	—
ϕ_1	1/3	—-	–	—
ϕ_2	1/3	—-	–	—

are set as suggested by their respective authors. T is the neighborhood size for MOEA/D and MOEA/D*, δ and n_r are the probability of selecting a determined neighborhood and the maximum number of replacements in the neighborhood (for MOEA/D*). P_c and P_m are the crossover rate and mutation rate. ω, ϕ_1 and ϕ_2 are the weights used in GPSO. Finally, the search for all the evolutionary approaches was restricted to perform 2,000 generations.

4.3 Performance Assessment

In this section, we outline the performance measures used in our comparison, and the method employed to define the reference set R.

4.3.1 Performance measures

Set Two Coverage ($\mathcal{C}$). *Set Two Coverage* ($\mathcal{C}$) was proposed by Zitzler et al. [36], and it compares a set of non-dominated solutions A with respect to another set B, using Pareto dominance. This performance measure is defined as:

$$\mathcal{C}(A, B) = \frac{|\{b \in B | \exists a \in A : a \succeq b\}|}{|B|} \tag{5}$$

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 (i.e. not always $\mathcal{C}(A, B) = \mathcal{C}(B, A)$ is held), we need to calculate both $\mathcal{C}(A, B)$ and $\mathcal{C}(B, A)$.

Inverted Generational Distance ($\mathcal{IGD}$). The Inverted Generational Distance ($\mathcal{IGD}$) [8] indicates how far a given Pareto front approximation is from a reference set. Let R be a proper representation of the Pareto optimal front, the $\mathcal{IGD}$ for a set of approximated solutions P is calculated as:

$$\mathcal{IGD}(P) = \frac{1}{|R|} \sum_{\mathbf{v} \in R} d(\mathbf{v}, P) \tag{6}$$

where $d(\mathbf{v}, P)$ is a minimum distance between $\mathbf{v}$ and any point in P and $|R|$ is the cardinality of R. The $\mathcal{IGD}$ metric can measure both convergence and diversity when the reference set R is a proper representation of the true Pareto front. A value of zero in this performance measure, indicates that all the solutions obtained by the algorithm are on the true Pareto front, it is the best possible value.

4.3.2 Reference set definition

From problem in Equation (2) and replacing the objective function and constraints by the multi-objective 1/0 knap-

sack problem (Equation (4)), we have:

$$\begin{aligned} \text{minimize:} \quad g^{tch}(\mathbf{x}|\lambda,\mathbf{z}) &= \max_{1\leq j\leq M}\{\lambda_j|z_j - f_j(\mathbf{x})|\} \\ &= \max_{1\leq j\leq M}\left\{\lambda_j\left|z_j - \left(\sum_{i=1}^{n} p_{ji}\cdot x_i\right)\right|\right\} \\ s.t. \quad \sum_{i=1}^{n} w_{ji}x_i &\leq c_j, \quad j=1,\ldots,M \end{aligned} \tag{7}$$

by introducing the relaxed formulation (i.e. allowing real numbers such that $0 \leq x_i \leq 1$), the above problem can be rewritten in its linear form as [12]:

$$\begin{aligned} \text{minimize:} \quad & \alpha \\ s.t. \quad & \lambda_j\left(z_j - \textstyle\sum_{i=1}^{n} p_{ji}x_i\right) \leq \alpha \\ & \textstyle\sum_{i=1}^{n} w_{ji}x_i \leq c_j, \\ & 0 \leq x_i \leq 1, \qquad i=1,\ldots,n \\ & \text{for all } j=1,\ldots,M \end{aligned} \tag{8}$$

where $\mathbf{z} = (z_1,\ldots,z_M)^\intercal$ is the reference point and $\lambda = (\lambda_1,\ldots,\lambda_M)^\intercal$ is a weight vector satisfying $\sum_{j=1}^{M} z_j = 1$ and $\lambda_i \geq 0$.

For each instance, we generated the reference set R solving a large number of relaxed linear problems (Equation (8)) defined by different weight vectors. Since the reference set R should contain a large enough number of points to ensure a good measurement of the $\mathcal{IGD}$ metric, we should generate a finite but well-distributed set of weight vectors. Simplex lattice method becomes impractical to define a specific number of weight vectors in high-dimensional spaces, therefore, we used the methodology presented in [32] and generated $200 \times M$ weight vectors (where M denotes the number of objectives). All the solutions for different weight vectors of problem (8), give the points $\mathbf{F}(\mathbf{x})$'s (in the objective space). Therefore, for a specific instance, the obtained points constitute the reference set R.[3]

For each instance, the reference point $\mathbf{z}$ was found by individual optimization of each separate objective in the relaxed multi-objective 0/1 knapsack problem. Note that all feasible solutions of the multi-objective 0/1 knapsack problem, are also feasible solutions of the relaxed problem. Therefore, the optimum values of each objective on the relaxed problem is not worse than the optimum value of this objective on the original problem [12][4].

5. DISCUSSION OF RESULTS

We compared experimentally our proposed MO-GPSO/D with MOEA/D, MOEA/D*, and NSGA-III on knapsack problems with 2, 3, 5, 8, and 10 objectives.

Figure 3 shows the results of the evaluation with the performance measure $\mathcal{C}$. In order to illustrate the general performance of the algorithms in comparison, simple box plots are shown. The thick line represents the median value, the upper and lower ends of the box are the upper and lower quartiles, and the ends of the vertical line are minimum and maximum values. We computed the $\mathcal{C}$ performance measure by comparing pairs of algorithms (i.e., $\mathcal{C}(A,B)$ and $\mathcal{C}(B,A)$). These values were obtained as average values of the comparisons of all independent runs of algorithm A with all independent runs of algorithm B.

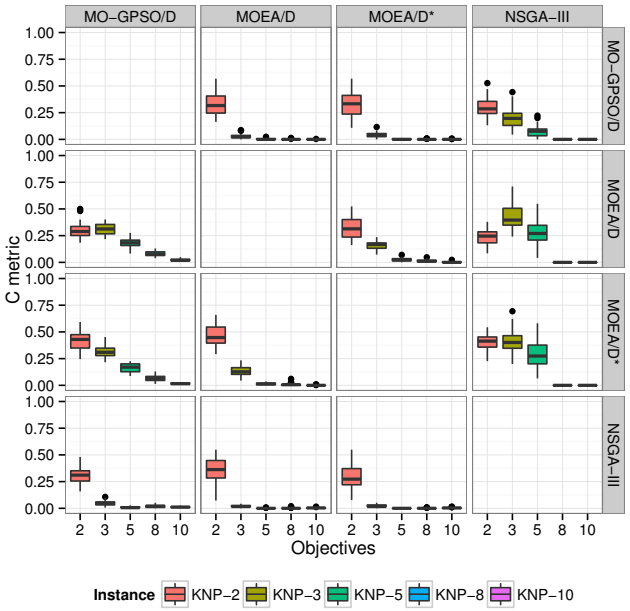

Figure 3: Results of the comparison with $\mathcal{C}(A,B)$ performance measure. Each chart contains five box plots representing the distribution of $\mathcal{C}$ values for a certain ordered pair of algorithms. Scale is zero at the bottom and one at the top for each chart. Chart in row of algorithm A and column of algorithm B presents values of convergence of approximations generated B by approximations generated by A.

In these charts, we show the ratio of solutions produced by MO-GPSO/D that dominate the solutions produced by MOEA/D, MOEA/D* and NSGA-III, respectively. This Figure must be read as $\mathcal{C}(A,B)$, where A denotes the algorithm in row, and B the algorithm in column. As we can see, the algorithms have a similar performance in the two-objective problem. The same behavior can be observed for problems with three objectives. However, when the number of objectives is increased, the ratio of solutions dominated by any algorithm decreases. It does not mean that the algorithms decrease their performance. When the values of $\mathcal{C}(A,B)$ and $\mathcal{C}(B,A)$ are almost the same, and they are small, it means that both algorithms are competitive in terms of Pareto relation (and this is the behavior of MOEAs with 5, 8, and 10 objectives). However, in terms of approximating the complete Pareto front, this metric could be misunderstood. It can be the case that algorithm A and B produce a small value for the $\mathcal{C}$ metric. However, solutions produced by algorithm A can draw a suitable representation of the Pareto front while solutions produced by algorithm B can be biased in a specific part of the Pareto front. In order to investigate precisely this behavior, we use the $\mathcal{IGD}$ performance measure which assesses the distance of solutions produced by an algorithm to the reference Pareto front.

Table 3 shows the results obtained by the algorithms in the comparison with respect to the second performance measure, $\mathcal{IGD}$. In each cell, the number on the top is the average indicator-value (lower is better), and the number be-

[3]The weight vectors and reference set for each instance are available at http://computacion.cs.cinvestav.mx/~zapoteca/MO-KNP/

[4]In order to solve the linear optimization problem, we use the Python SciPy library by employing the Simplex method.

Table 3: Table of results achieved by MO-GPSO/D, MOEA/D, MOEA/D*, and NSGA-III for the MO-KNP with 2, 3, 5, 8, and 10 objectives in the IGD performance measure.

M	MO-GPSO/D	MOEA/D	MOEA/D*	NSGA-III
	$\mathcal{IGD}$ (σ)	$\mathcal{IGD}$ (σ)	$\mathcal{IGD}$ (σ)	$\mathcal{IGD}$ (σ)
2	**31.30836** (3.661)	42.88855 (4.873)	44.22541 (5.915)	43.32292 (7.194)
3	**83.43913** (5.226)	106.15929 (8.826)	107.29307 (9.350)	174.16725 (13.750)
5	**302.59052** (6.888)	404.47921 (14.779)	399.88836 (17.213)	619.17796 (18.129)
8	518.70840 (11.896)	507.04133 (7.716)	**505.50804** (7.515)	847.17804 (19.274)
10	**526.08313** (5.417)	534.27874 (7.710)	531.78527 (6.005)	868.40125 (22.892)

low it in brackets is the standard deviation. Best values for each problem instance are reported in bold. Underlined values correspond to algorithms that are *not* statistically outperformed by any other algorithm for the instance under consideration (using a Mann-Whitney-Wilcoxon [30] non-parametric statistical test with a p-value of 0.05 with Bonferroni correction [4]).

As we can see from Table 3, the performance of the MOEA/D, MOEA/D* and NSGA-III are very similar for instances with two objectives, while MO-GPSO/D is significantly better. The performance of MO-GPSO/D is (statistically significantly) better than all the other algorithms in the comparison, except on the problem with 8 objectives, on which MOEA/D* is better than MO-GPSO/D, but the differences between their performance is not statistically significant. This analysis suggests that the way we couple GPSO into a multi-objective decomposition framework is a good strategy for the type of problems under study. It is also remarkable that the performance of NSGA-III decreases as the number of objective increases. The main reason for this is that NSGA-III relies on a suitable construction of the hyperplane, which is essential for a correct fitness assignment to solutions. Such hyperplane is defined by finding the best solutions in the population that minimizes the achievement scalarization function (ASF) [9] with the canonical basis (in $\mathbb{R}^M$). However, in discrete problems, optimal solutions to the ASF with the specific weight vector cannot be found (it could be not exist). This could generate a bad definition of such hyperplane and lead to a wrong ranking of solutions.

Finally, Figure 4 reports the complete convergence plots for the algorithms in the comparison. This plot also corroborates the good performance achieved by MO-GPSO/D in the test instances (specially in problems with 2, 3, and 5 objectives), as the plot of MO-GPSO/D is lower than the plots of the other algorithms.

6. CONCLUSIONS AND FUTURE WORK

We have proposed a new multi-objective particle swarm optimizer using the geometric particle swarm optimization algorithm within a multi-objective framework based on decomposition. The proposed approach was designed to deal with combinatorial MOPs with both low and high dimensionality (in terms of the number of objectives).

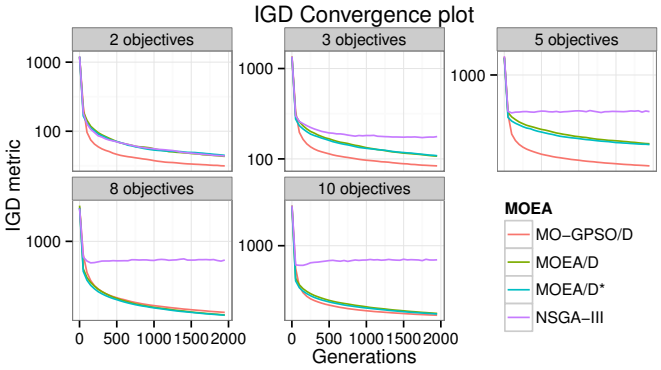

Figure 4: Convergence plots (in log scale) for $\mathcal{IGD}$ performance measure on the MO-KNPs having 2, 3, 5, 8, and 10 objective functions, respectively.

The proposed approach follows the decomposition approach in the sense that it optimizes a set of scalarizing functions but it does not follow other principles of MOEA/D, i.e. neighborhoods, dynamic selection or limit on maximum number replacements.

Experimental results indicate that our proposed approach (i.e. MO-GPSO/D) outperforms significantly three state-of-the-art MOEAs based on decomposition, namely MOEA/D, MOEA/D*, and NSGA-III, on the test problems adopted.

In future work, we will test MO-GPSO/D on more complex and on a greater variety of problems to identify strengths and weakness of this algorithm. We will also analyze the scalability of MO-GPSO/D with MOPs with large scale i.e., large number of bits. Finally, we will consider improvements to the proposed approach by introducing local search mechanism during the search process.

7. REFERENCES

[1] D. Agrafiotis and W. CedeÃśo. Feature selection for structure-activity correlation using binary particle swarms. *Journal of Medicinal Chemistry*, 45(5):1098–1107, 2002.

[2] E. Alba, J. García-Nieto, L. Jourdan, and E.-G. Talbi. Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 284–290. IEEE, 2007.

[3] E. Alba, J. García-Nieto, J. Taheri, and A. Zomaya. New research in nature inspired algorithms for mobility management in gsm networks. In *Applications of Evolutionary Computing*, pages 1–10. Springer, 2008.

[4] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

[5] V. J. Bowman Jr. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Lecture Notes in Economics and Mathematical System*, volume 130, pages 76–86. 1976.

[6] I. Chaman Garcia, C. Coello Coello, and A. Arias-Montano. Mopsohv: A new hypervolume-based multi-objective particle swarm

optimizer. In *CEC'2014*, pages 266–273, July 2014.

[7] M. Clerc. Discrete particle swarm optimization, illustrated by the traveling salesman problem. In *New Optimization Techniques in Engineering*, pages 219–239. Springer, 2004.

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

[9] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE TEVC*, 18(4):577–601, August 2014.

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

[11] J. García-Nieto and E. Alba. Parallel multi-swarm optimizer for gene selection in dna microarrays. *Applied Intelligence*, 37(2):255–266, 2012.

[12] A. Jaszkiewicz. On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem—A Comparative Experiment. *IEEE TEVC*, 6(4):402–412, August 2002.

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

[14] J. Kennedy and R. C. Eberhart. A discrete binary version of the particle swarm algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 5:4104–4108, 1997.

[15] F. Li, J. Liu, S. Tan, and X. Yu. R2-mopso: A multi-objective particle swarm optimizer based on r2-indicator and decomposition. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 3148–3155, May 2015.

[16] H. Li and Q. Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE TEVC*, 13(2):284–302, April 2009.

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

[18] C. Mohan and B. Al-Kazemi. Discrete particle swarm optimization. In *workshop on particle swarm optimization*, Indianapolis, 2001. Purdue School of Engineering and Technology, IUPUI.

[19] A. Moraglio, C. D. Chio, and R. Poli. Geometric particle swarm optimization. In *European Conference on Genetic Programming*, pages 125–136, 2007.

[20] A. Moraglio and J. Togelius. Geometric pso for the sudoku puzzle. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 118–125, 2007.

[21] 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.

[22] G. Pampara, A. Engelbrecht, and N. Franken. Binary differential evolution. In *IEEE Congress on Evolutionary Computation*, 2006.

[23] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *GrC'2008*, pages 534 –537, 2008.

[24] R. Purshouse and P. Fleming. On the evolutionary optimization of many conflicting objectives. *Evolutionary Computation, IEEE Transactions on*, 11(6):770–784, Dec 2007.

[25] J. Qin, Y.-x. Yin, and X.-j. Ban. Hybrid discrete particle swarm algorithm for graph coloring problem. *Journal of Computers*, 6(6):1175–1182, 2011.

[26] 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.

[27] H. Scheffé. Experiments With Mixtures. *Journal of the Royal Statistical Society*, Series B (Methodological), 20(2):344–360, 1958.

[28] E.-G. Talbi, L. Jourdan, J. Garcia-Nieto, and E. Alba. Comparison of population based metaheuristics for feature selection: Application to microarray data classification. In *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*, pages 45–52. IEEE, 2008.

[29] J. Togelius, R. D. Nardi, and A. Moraglio. Geometric pso + gp = particle swarm programming. In *Proceedings of the Congress on Evolutionary Comptutation (CEC)*, 2008.

[30] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[31] Y. yan Tan, Y. chang Jiao, H. Li, and X. kuan Wang. Moea/d + uniform design: A new version of moea/d for optimization problems with many objectives. *Computers & Operations Research*, 40(6):1648–1660, 2013.

[32] S. Zapotecas-Martínez, H. E. Aguirre, K. Tanaka, and C. A. Coello Coello. On the Low-Dyscrepancy Sequences and Their use in MOEA/D for High-Dimensional Objective Spaces. In *CEC'2015*, pages 2835–2842, Sendai, Japan, May 2015.

[33] S. Zapotecas Martínez and C. A. Coello Coello. A Multi-objective Particle Swarm Optimizer Based on Decomposition. In *GECCO'2011*, pages 69–76, Dublin, Ireland, July 2011. ACM Press.

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

[35] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, University of Essex and Nanyang Technological University, 2008.

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

[37] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE TEVC*, 3(4):257–271, November 1999.