VSD-MOEA: A Dominance-Based Multi-Objective Evolutionary Algorithm with Explicit Variable Space Diversity Management

Joel Chacón Castillo joel.chacon@cimat.mx
Computer Science Department, Center for Research in Mathematics (CIMAT),
Callejón Jalisco s/n, Mineral de Valenciana, Guanajuato, Guanajuato 36240, Mexico

Carlos Segura carlos.segura@cimat.mx
Computer Science Department, Center for Research in Mathematics (CIMAT),
Callejón Jalisco s/n, Mineral de Valenciana, Guanajuato, Guanajuato 36240, Mexico

Carlos A. Coello Coello ccoello@cs.cinvestav.mx
Computer Science Department, Center for Research and Advanced Studies (CINVESTAV-IPN), Mexico City 07300

Abstract

Most state-of-the-art Multi-Objective Evolutionary Algorithms (MOEAs) promote the preservation of diversity of objective function space but neglects the diversity of decision variable space. The aim of this paper is to show that controlling explicitly the amount of diversity maintained in the decision variable space is useful to increase the quality of MOEAs when taking into account metrics of the objective space. Our novel Variable Space Diversity based MOEA (VSD-MOEA) explicitly considers the diversity of both decision variable and objective function space. This information is used with the aim of properly adapting the balance between exploration and intensification during the optimization process. Particularly, at the initial stages, decisions made by the approach are more biased by the information on the diversity of the variable space, whereas it gradually grants more importance to the diversity of objective function space as the evolution progresses. The latter is achieved through a novel density estimator. The new method is compared with state-of-art MOEAs using several benchmarks with two and three objectives. This novel proposal yields much better results than state-of-the-art schemes when considering metrics applied on objective function space, exhibiting a more stable and robust behavior.

Keywords

Multi-objective Evolutionary Algorithms, Premature Convergence, Diversity Preservation

1 Introduction

Multi-objective Optimization Problems (MOPs) involve the simultaneous optimization of several objective functions that are usually in conflict with each other (Deb and Kalyanmoy, 2001). A continuous box-constrained minimization MOP, which is the kind of problem addressed in this paper, can be defined as follows:

$$\begin{aligned} &minimize \quad \vec{F} = [f_1(\vec{\mathbf{x}}), f_2(\vec{\mathbf{x}}), ..., f_M(\vec{\mathbf{x}})] \\ &subject \quad to \quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, ..., n. \end{aligned} \tag{1}$$

where n corresponds to the dimensionality of the decision variable space, $\vec{\mathbf{x}}$ is a vector of n decision variables $\vec{\mathbf{x}} = (x_1, ..., x_n) \in \mathbb{R}^n$, which are constrained by $x_i^{(L)}$ and $x_i^{(U)}$, i.e. the lower bound and the upper bound, and M is the number of objective functions to optimize. The feasible space bounded by $x_i^{(L)}$ and $x_i^{(U)}$ is denoted by Ω. Each solution is mapped to the objective space with the function $F : \Omega \to \mathbb{R}^M$, which consists of M real-valued objective functions, and $\mathbb{R}^M$ is called the *objective space*.

Given two solutions $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \Omega$, $\vec{\mathbf{x}}$ dominates $\vec{\mathbf{y}}$, which is mathematically denoted by $\vec{\mathbf{x}} \prec \vec{\mathbf{y}}$, iff $\forall m \in 1, 2, ..., M : f_m(\vec{\mathbf{x}}) \leq f_m(\vec{\mathbf{y}})$ and $\exists m \in 1, 2, ..., M : f_m(\vec{\mathbf{x}}) < f_m(\vec{\mathbf{y}})$. The best solutions of an MOP are those that are not dominated by any other feasible vector. These solutions are known as the Pareto optimal solutions. The Pareto set is the set of all Pareto optimal solutions, and the Pareto front is the image (i.e., the corresponding objective function values) of the Pareto optimal set. The goal of multi-objective optimizers is to obtain a proper approximation of the Pareto front, i.e. a set of well-distributed solutions that are close to the Pareto front.

One of the most popular meta-heuristics used to deal with MOPs is the Evolutionary Algorithm (EA). In single-objective EAs, it has been shown that taking into account the diversity of decision variable space to properly balance between exploration and exploitation is highly important to attain high quality solutions (Herrera et al., 1996). Diversity can be taken into account in the design of several components, such as in the variation stage (Herrera and Lozano, 2003; Mitchell, 1998), the replacement phase (Segura et al., 2016) and/or the population model (Koumousis and Katsaras, 2006). The explicit consideration of diversity leads to improvements in terms of avoiding premature convergence, meaning that taking into account diversity in the design of EAs is specially important when dealing with long-term executions. Recently, some diversity management algorithms that combine the information on diversity, stopping criterion and elapsed generations have been devised. They have yielded a gradual loss of diversity that depends on the time or evaluations granted to the execution (Segura et al., 2016). Specifically, the aim of such a methodology is to promote exploration in the initial generations and gradually alter the behavior towards intensification. These schemes have provided highly promising results. For instance, new best-known solutions for some well-known variants of the frequency assignment problem (Segura et al., 2017), and for a two-dimensional packing problem (Segura et al., 2016) have been attained using the same principles. Additionally, this principle guided the design of the winning strategy of the Second Wind Farm Layout Optimization Competition[1], which was held at the *Genetic and Evolutionary Computation Conference* and of the extended round of Google Hash Code 2020 [2] , with more than $100,000$ participants. Thus, the benefits of this type of design patterns have been shown in several different single-objective optimization problems.

One of the goals when designing Multi-objective Evolutionary Algorithms (MOEAs) is to obtain a well-spread set of solutions in objective function space. As a result, most state-of-the-art MOEAs consider the diversity of the objective space explicitly. However, this is not the case for the diversity of decision variable space. Maintaining some degree of diversity in objective space implies that full convergence is not achieved in decision variable space (Kukkonen and Lampinen, 2009). In some way, decision variable space inherits some degree of diversity due to the way in which objective space is taken into account. However, this is just an indirect way of preserving diversity of deci-

[1]https://www.irit.fr/wind-competition/
[2]https://codingcompetitions.withgoogle.com/hashcode/

sion variable space, so in some cases the level of diversity might not be large enough to ensure a proper degree of exploration. For instance, it has been shown that with some of the WFG test problems, in most state-of-the-art MOEAs the *distance parameters* quickly converge, meaning that the approach focuses just on optimizing the *position parameters* for a long period of the optimization process (Castillo et al., 2017). Thus, while some degree of diversity is maintained, a situation similar to premature convergence occurs, meaning that genetic operators might not be longer able to generate better trade-offs.

In light of the differences between state-of-the-art single-objective EAs and MOEAs, this paper proposes a novel MOEA, the Variable Space Diversity based MOEA (VSD-MOEA), which relies on explicitly controlling the amount of diversity in decision variable space. Similarly to the successful methodology applied in single-objective optimization, the stopping criterion and the number of evaluations performed are used to vary the amount of diversity desired in decision variable space. The main difference with respect to the single-objective case is that diversity of the objective function space is simultaneously considered by using a novel objective space density estimator. Particularly, the approach grants more importance to the diversity of decision variable space in the initial stages, and it gradually grants more importance to the diversity of objective function space as the evolution progresses. In fact, in the last stage of execution, diversity of decision variable space is neglected. Thus, in the last phases, the proposed approach is quite similar to current state-of-the-art approaches. To the best of our knowledge, this is the first MOEA whose design follows this adaptive principle.

Since there currently exists quite a large number of different MOEAs (Tan et al., 2005), three popular schemes have been selected to validate our proposal. This validation was performed using several well-known benchmarks and proper quality metrics. This paper clearly shows the important benefits of properly taking into account the diversity of decision variable space. In particular, the advantages become more evident in the most complex problems. Note that this is consistent with the single-objective case, where the most important benefits have been obtained in complex multi-modal cases (Segura et al., 2017). It is also important to clarify that, in spite of considering the variable-space diversity, our work is not a niche-based proposal for multimodal optimization (Deb and Tiwari (2005), Zhou et al. (2009), Li et al. (2016),Liang et al. (2016)). Instead, this work is oriented to show that controlling explicitly the amount of diversity maintained in the decision variable space is useful to increase the quality of MOEAs when taking into account metrics of the objective space.

The rest of this paper is organized as follows. Section 2 provides a review of the previous related work. Additionally, some key components related to diversity and to the VSD-MOEA design are discussed. The VSD-MOEA proposal is detailed in Section 3. Section 4 is devoted to the experimental validation of the proposal. Finally, our conclusions and some lines of future work are given in Section 5. Note also that some supplementary materials are also provided. They include details of the experimental results with additional performance measures, as well as an explanatory video.

2 Literature Review

This section is devoted to reviewing some of the most relevant work which is closely related to our proposal. First, some of the most popular ways of managing diversity in EAs are presented. Then, the state of the art in MOEAs is summarized.

2.1 Diversity Management in Evolutionary Algorithms

The proper balance between exploration and exploitation is one of the keys to designing successful EAs. In the single-objective domain, it is known that properly managing the diversity of decision variable space is a way to achieve this balance, and as a consequence, a large number of diversity management techniques have been devised (Pandey et al., 2014). Specifically, these methods are classified depending on the component(s) of the EA that is modified to alter the way in which diversity is maintained. A popular taxonomy identifies the following groups (Črepinšek et al., 2013): *selection-based*, *population-based*, *crossover/mutation-based*, *fitness-based*, and *replacement-based*. Additionally, the methods are referred to as *uniprocess-driven* when a single component is altered, whereas the term *multiprocess-driven* is used to refer to those methods that act on more than one component.

Among the previous proposals, the replacement-based methods have yielded very high-quality results in recent years (Segura et al., 2017), so this alternative was selected with the aim of designing a novel MOEA that explicitly incorporates a way to control the diversity of decision variable space. The basic principle of these methods is to bias the level of exploration in successive generations by controlling the diversity of the survivors. Since premature convergence is one of the most common drawbacks in the application of EAs, modifications are usually performed with the aim of slowing down convergence. One of the most popular proposals belonging to this group is the *crowding* method, which is based on the principle that offspring should replace similar individuals to those from the previous generation (Mengshoel et al., 2014). Several replacement strategies that do not rely on crowding have also been devised. In some methods, diversity is considered as an objective. For instance, in the hybrid genetic search with adaptive diversity control (HGSADC) proposed by Vidal et al. (2013), individuals are sorted by their contribution to diversity and by their original cost. Then, the rankings of the individuals are used in the fitness assignment phase. A more recent proposal (Segura et al., 2017) incorporates a penalty approach to gradually alter the amount of diversity maintained in the population. Specifically, the initial phases preserve a higher amount of diversity than the final phases of the optimization. This last method has inspired the design of the novel proposal put forth in this paper for multi-objective optimization.

It is important to remark that in the case of multi-objective optimization, little work related to maintaining the diversity of decision variable space has been done. The following section reviews some of the most important MOEAs and introduces some of the works that consider the maintenance of diversity of decision variable space.

2.2 Multi-objective Evolutionary Algorithms

In recent decades, several MOEAs have been proposed. While most of them seek to provide a well-spread set of solutions close to the Pareto front, several ways of achieving this purpose have been devised. Therefore, several taxonomies have been proposed with the aim of better classifying the different schemes (Bechikh et al., 2016). Particularly, a MOEA can be designed based on Pareto dominance, indicators and/or decomposition (Trivedi et al., 2016). Since none of the groups is significantly superior to the others, in this work all of them are taken into account to validate our proposal. This section introduces the three types of schemes and some of the most popular approaches belonging to each category. Then, one MOEA in each category is selected to validate the VSD-MOEA.

The dominance-based category includes those schemes where the Pareto dominance relationship is used to guide the design of some of its components, such as the fitness assignment, parent selection and replacement phase. The dominance relationship does not inherently promote the preservation of diversity in objective function space; therefore, additional techniques such as objective space density estimators are usually integrated in order to obtain a proper spread and convergence to the Pareto front. The most popular dominance-based MOEA is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) proposed by Deb et al. (2002).

Several quality indicators have been devised to assess the performance of MOEAs. In indicator-based MOEAs, the use of the Pareto dominance relationship is substituted by some quality indicators to guide the decisions made by the MOEA. An advantage of this kind of algorithm is that the indicators usually take into account both the quality and the diversity of objective function space. Thus, incorporating additional mechanisms to promote diversity in objective function space is not required. The Indicator-Based Evolutionary Algorithm (IBEA) proposed by Zitzler and Künzli (2004) was the first method belonging to this category. A more recent one is the R2-Indicator-Based Evolutionary Multi-objective Algorithm (R2-EMOA) proposed by Trautmann et al. (2013), whose performance in MOPs has been quite promising. Its most important feature is the use of the R2 indicator.

Finally, decomposition-based MOEAs (Chiang and Lai, 2011) are based on transforming the MOP into a set of single-objective optimization problems that are tackled simultaneously. This transformation can be performed in several ways, e.g. with a linear weighted sum or with a weighted Tchebycheff function. Given a set of weights to establish different single-objective functions, the MOEA searches for a single high-quality solution for each of them. The weight vectors should be selected with the aim of obtaining a well-spread set of solutions (Deb and Kalyanmoy, 2001). The Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) proposed by Zhang and Li (2007) is the most popular decomposition-based MOEA. Its main principles include problem decomposition, weighted aggregation of objectives and mating restrictions through the use of neighborhoods.

It is important to note that none of the most popular algorithms in the multi-objective field introduces special mechanisms to promote diversity of decision variable space. However, some efforts have been devoted to this principle. A popular approach to promote the diversity of decision variable space is the application of fitness sharing (Horn et al., 1994), in a way similar to single-objective optimization. Although fitness sharing might be used to promote the diversity of both objective and decision variable spaces, most popular variants consider only distances in objective function space. Another MOEA designed to promote diversity of both decision variable space and objective function space is the Genetic Diversity Evolutionary Algorithm (GDEA) proposed by Toffolo and Benini (2003). In this case, each individual is assigned a diversity-based objective which is calculated as the Euclidean distance in the genotype space to the remaining individuals in the population. Then, a ranking that considers both the original objectives and the diversity objective is used to sort individuals. Another somewhat popular approach is to calculate distances between candidate solutions by taking into account both objective and decision variable space (Deb and Tiwari, 2005; Shir et al., 2009) with the aim of promoting diversity of both spaces. A different proposal combines the use of two selection operators (Chan and Ray, 2005). The first one promotes diversity and quality in objective function space, whereas the second one promotes diversity in decision variable space. A different approach involves modifying the hypervolume to integrate the decision variable space diversity into a single

Algorithm 1 Main procedure of VSD-MOEA

1: **Initialization**: Generate an initial population P_0 with N individuals.
2: **Evaluation**: Evaluate all individuals in the population.
3: Assign $t = 0$
4: **while** (not stopping criterion) **do**
5: **Mating selection**: Fill the mating pool by performing binary tournament selection on P_t, based on the non-dominated ranks (ties are broken randomly).
6: **Variation**: Apply SBX and Polynomial-based mutation to the mating pool to create an offspring population Q_t with N individuals.
7: **Evaluation**: Evaluate all individuals in Q_t.
8: **Survivor selection**: Generate P_{t+1} by applying the replacement scheme described in Algorithm 2, using P_t and Q_t as inputs.
9: $t = t + 1$

metric (Ulrich et al., 2010). In this approach, the proposed metric is used to guide the selection in the MOEA. Finally, some indirect mechanisms that might affect diversity have also been taken into account. The most popular one is probably the use of mating restrictions (Ishibuchi and Shibata, 2003; Chiang and Lai, 2011).

In light of the results of the approaches described above, it is clear that considering the diversity of decision variable space in the design phase might yield benefits for decision makers, since the final solutions obtained by these methods exhibit a higher decision variable space diversity than those obtained by traditional approaches (Deb and Tiwari, 2005; Rudolph et al., 2007). Thus, while clear improvements are obtained when metrics related to the decision variable space are taken into account, the benefits in terms of the objective function space are not so clear. We claim that one of the reasons for this behavior might be that the diversity of decision variable space is considered in the whole optimization process. However, in a similar way as in the single objective domain, reducing the importance granted to the diversity of decision variable space as the generations progress (Segura et al., 2016) might be truly important for obtaining better approximations of the Pareto front. Currently, no MOEA considers this idea, therefore this principle has guided the design of our novel MOEA.

3 Proposal

This section provides a full description of our proposal called *Variable Space Diversity based MOEA* (VSD-MOEA) [3]. The novelty of VSD-MOEA appears in the replacement phase, which incorporates the use of variable space diversity and a novel objective space density estimator. The main principle behind the design of the novel replacement is to use the stopping criterion and elapsed generations with the aim of gradually moving from exploration to exploitation during the search process. Note that this principle might be incorporated in any of the three categories of MOEAs. In this paper, our decision was to incorporate it into a dominance-based approach. Note that this category has been particularly suitable for problems with two and three objectives. Thus, some of our design decisions might not be suitable for dealing with many-objective optimization problems.

The general framework of VSD-MOEA is quite standard. Algorithm 1 shows the pseudo-code of VSD-MOEA. Parents are selected using a binary tournament based on dominance ranking with ties broken randomly. The variation stage is based on applying the well-known Simulated Binary Crossover (SBX) and polynomial-based mu-

[3] The source code in C++ is freely available at https://github.com/carlossegurag/VSD-MOEA

tation (Agrawal et al., 1994; Deb and Goyal, 1996) operators. Thus, the contribution appears in the replacement phase. The rest of this section is devoted to describing the replacement phase, including the novel objective space density estimator.

3.1 Replacement Phase of VSD-MOEA

The replacement phase of EAs is in charge of deciding, for each generation, which members of the previous population together with their corresponding offspring will survive. The novel replacement scheme promotes a gradual movement from exploration to exploitation, which has been a highly beneficial principle in the design of single-objective optimizers (Segura et al., 2016). Specifically, the replacement phase operates as follows. First, the members of the previous population and offspring are merged in a multi-set with $2 \times N$ individuals. Then, an iterative process that selects an additional individual at each iteration is used to pick the N survivors. In order to take into account the diversity of decision variable space, the Distance to Closest Survivor (DCS) of each individual is calculated at each iteration. Thus, the DCS of an individual I is calculated as $\min_{s \in S} Distance(I, s)$, where S is the multi-set containing the currently selected survivors. Normalized Euclidean distances are considered, so in order to calculate distances between any two individuals A and B, Eq. (2) is applied. In the first iteration, the S multi-set is empty, so the DCS of each individual is infinity.

$$Distance(A, B) = \left(\frac{1}{n} \sum_{i=1}^{n} \left(\frac{A_i - B_i}{x_i^{(U)} - x_i^{(L)}} \right)^2 \right)^{1/2} \tag{2}$$

Note that individuals with larger DCS values are those that contribute more significantly to promoting exploration. In order to avoid an excessive decrease in the degree of exploration, individuals with a DCS value below a certain threshold are penalized. Then, among the non-penalized individuals, an objective space density estimator is used to select the additional survivor of the iteration. In our case, the novel density estimator described in the next subsection is used. Note that it might happen that all individuals are penalized, in which case the individual with the largest DCS is selected to survive.

In order to better understand the penalty method, it can be visualized in the following way. After selecting each survivor, a hyper-sphere centered around a candidate solution — in decision variable space — is created. Then, all the individuals that are inside the hyper-sphere are penalized, with the objective space density estimator only taking into account the survivors and the non-penalized individuals. This is illustrated in Fig. 1, which represents a state where three individuals have been selected to survive and an additional survivor must be picked. The left side shows individuals in decision variable space. Current survivors are marked with a red border. Each of them is surrounded by a dashed blue circle of radius D_t. In this scenario, the penalized individuals are numbers 4, 5, and 6. In objective function space — right side — the penalized individuals are shown in gray, indicating that the objective space density estimator is not considering them.

Since using a large radius for the hyper-spheres induces a large degree of exploration, it makes sense to alter this value during the optimization process. This is precisely one of the keys of our proposal. The sizes of the hyper-spheres are modified dynamically by taking into account the stopping criterion and elapsed generations. Specifically, the radius is decreased linearly starting from an initial distance.

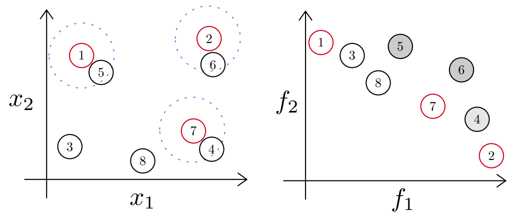

Figure 1: Penalty Method of the Replacement Phase - The left side represents decision variable space and the right side represents objective function space.

Algorithm 2 Replacement Phase of VSD-MOEA

1: Input: P_t (Population of current generation), Q_t (Offspring of current Generation), N (Population Size)
2: Output: P_{t+1}
3: $R_t = P_t \cup Q_t$
4: $P_{t+1} = \emptyset$
5: $Penalized = \emptyset$
6: $D_t = D_I - D_I * \frac{G_{Elapsed}}{0.5 * G_{End}}$
7: **while** $|P_{t+1}| \leq$ N **do**
8: Compute DCS of individuals in R_t, using P_{t+1} as a reference set
9: Move the individuals in R_t with $DCS < D_t$ to $Penalized$
10: **if** R_t is empty **then**
11: Compute DCS of individuals in $Penalized$, using P_{t+1} as a reference set
12: Move the individual in $Penalized$ with the largest DCS to R_t
13: Identify the first front (F) in $R_t \cup P_{t+1}$ with an individual $I \in R_t$
14: Use the novel density estimator (Algorithm 3) to select a new survivor from F and move it to P_{t+1}
15: **return** P_{t+1}

This means that in the initial phases, exploration is promoted. However, as the size of the radius decreases, only very close individuals are penalized, meaning that more exploitation is allowed. Note that this method requires a parameter that is the initial radius of the hyper-spheres or initial threshold value. This parameter is denoted by D_I. Assigning a large value to this parameter might result in many individuals being penalized, which might thus maintain non-useful diversity. However, a value that is too small might not prevent fast convergence, meaning the approach might behave as a traditional non-diversity based MOEA. The robustness of the proposal with respect to this additional parameter is studied in our experimental validation.

Algorithm 2 formalizes the replacement phase of VSD-MOEA. First, the population of the previous generation (P_t) and the offspring (Q_t) are merged in R_t (line 3). At each iteration, the multi-set R_t contains the remaining non-penalized individuals that might be selected to survive. The population of survivors (P_{t+1}) and the set containing the penalized individuals are initialized to the empty set (lines 4 and 5). Then, the threshold value (D_t) that is used to penalize individuals that are too close is calculated (line 6). Note that D_I denotes the initial threshold value, $G_{Elapsed}$ is the number of generations that have evolved, and G_{End} is the stopping criterion, i.e., the number of generations that are to be evolved during the execution of the VSD-MOEA. The linear decrease is calculated such that after 50% of the total number of generations, the D_t value is below 0, meaning that no penalties are applied. This means that in the first 50% of the generations, more exploration is induced than in traditional MOEAs.

Algorithm 3 Density estimator

1: Input: P_{t+1} (Survivors), R_t (Candidates), F (Current front)
2: Output: $I \in R_t$
3: $FP = P_{t+1} \cap F$
4: $FR = R_t \cap F$
5: **for** $k \in$ number of objectives **do**
6:     Select the best individual $I \in F$ of k according to Eq. 3.
7:     **if** $I \in FR$ **then**
8:         **return** I
9: $MaxID = 0$
10: **for** $Ic \in FR$ **do**
11:     $Improvement = \min_{s \in FP} ID(Ic, s)$
12:     **if** $Improvement > MaxID$ **then**
13:         $MaxID = Improvement$
14:         $I = Ic$
15: **return** I

Then, an iterative process that selects an individual in each iteration is executed until the survivor set contains N individuals (line 7). The iterative process works as follows. First, the DCS value of each remaining non-penalized individual is calculated (line 8). Then, those individuals with a DCS value lower than D_t are moved to the set of penalized individuals (line 9). If all the remaining individuals are penalized (line 10), it means that the amount of exploration is lower than desired. Thus, the individual with the largest DCS value is recovered, i.e., moved to the set of non-penalized individuals (lines 11 and 12), and thus survives. Finally, the objective function space is considered. Specifically, candidate non-penalized individuals and current survivors are merged. Then, the well-known non-dominated sorting procedure proposed in Deb et al. (2002) is executed on this set, stopping as soon as a front with at least one candidate individual is found, i.e. with an individual of R_t (line 13). Then, taking the identified front as an input, a novel objective space density estimator is used to select the next survivor (line 14). The specific way in which each individual's contribution to the diversity of the objective space is measured is described in the next section.

3.2 A Novel Density Estimator for Objective Function Space

Since the dominance definition is not related to the preservation of diversity in objective function space, dominance-based MOEAs usually incorporate objective-space density estimators to promote the survival of diverse individuals. As previously described, our density estimator selects a new survivor from the front identified in line 13 of Algorithm 2. This front (referred in Algorithm 3 as F) contains at least one individual belonging to R_t, and it might also contain some elements of P_{t+1}. The aim behind the selection of the next survivor is to pick an individual of the input front that contributes significantly in terms of the quality and diversity of the objective space.

Algorithm 3 describes the selection of the next survivor. First, the sets FP and FR are identified (lines 3 and 4). FP contains the current survivors that are in F (already selected to the next generation P_{t+1}), whereas FR contains the remaining non-penalized individuals that are in F. Then, similarly to most state-of-the-art algorithms, a step to promote the selection of boundary solutions is included. Note that selecting the best solution for each objective might cause some drawbacks related to accepting a small improvement in one objective at the expense of significant degradation in other objectives. This issue can be solved by applying augmented functions (Deb and

Abouhawwash, 2016), which was our design choice. Specifically, for each objective k, the candidate solution that minimizes the Augmented Function (AF) given in Eq. 3 is iteratively identified (lines 5 to 8). If this individual belongs to FR, i.e., it has not yet been selected as a survivor, the next survivor is such an individual and the process ends (line 8). Note that augmented functions usually take into account weight vectors in order to deal with objectives that exhibit very different scales. Since benchmarks that have similar scales in each objective have been used in this paper, there was no need to apply the aforementioned weight vectors.

$$AF_k(\vec{x}) = f_k(\vec{x}) + 10^{-4} \times \sum_{j=1}^{M} f_j(\vec{x}) \tag{3}$$

In cases where the individuals that optimize each AF_K function are already in P_{t+1}, a contribution to objective-space diversity and quality is calculated for each individual in FR (lines 9 to 15). This contribution is calculated by taking into account the current survivors of the front (FP). Specifically, the "Improvement Distance" (ID) defined for the indicator IGD+ (Ishibuchi et al., 2015) is used. The ID of an individual A with respect to an individual B is calculated by taking into account only the objectives where A is better. Specifically, Eq. (4) is used.

$$ID(A, B) = \left(\sum_{i=1}^{M} (max(0, f_i(B) - f_i(A)))^2 \right)^{1/2} \tag{4}$$

The contribution of each member in FR (I) is calculated as $\min_{s \in FP} ID(I, s)$. Then, the individual with the highest contribution is selected as the next survivor (lines 12 to 14).

4 Experimental Validation

This section describes the experimental validation carried out to study the performance and gain a clear understanding of the specifics of our proposed VSD-MOEA. Our results clearly show that controlling the diversity of decision variable space provides a way to further improve the results obtained by state-of-art MOEAs. First, we discuss some technical specifications involving the benchmark problems and algorithms implemented. We then present a comparison between VSD-MOEA and state-of-the-art algorithms in long-term executions. Then, three additional experiments to fully validate VSD-MOEA are included. These analyses are designed to test the scalability in decision variable space, the performance with different stopping criteria, and the behavior with different initial penalty thresholds.

This work takes into account some of the most popular and widely used benchmarks in the multi-objective field. These problems are the WFG (Huband et al., 2005a), DTLZ (Deb et al., 2005), and UF (Zhang et al., 2008) test suites configured in a standard way. The WFG test problems were used with two and three objectives and were configured with 24 parameters, 20 of them corresponding to distance parameters and 4 to position parameters. In the DTLZ test problems, the number of variables was set to $n = M + r - 1$, where $r = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7, respectively. The UF benchmark comprises seven problems with two objectives (UF1-7) and three problems with three objectives (UF8-10). All of them were configured with

Table 1: Parameterization of the crossover probability applied in each MOEA

	2 objectives	3 objectives
VSD-MOEA	0.4	0.4
CPDEA	1.0	0.6
MOEA/D	1.0	1.0
R2-EMOA	1.0	0.2

Table 2: Parameterization applied to each MOEA

Algorithm	Configuration
MOEA/D	Max. updates by sub-problem (η_r) = 2, tour selection = 10, neighbor size = 10, period utility updating = 30 generations, local selection probability (δ) = 0.9,
VSD-MOEA	Initial threshold value (D_I) = 0.4
R2-EMOA	Equally distributed weight vectors (ρ) = 1, offspring by iteration = 1
CPDEA	Nearest neighbors (K) = 3, weight of standard deviation (η) = 2

30 variables. Note that the experiment used to analyze scalability considers different numbers of variables.

The experimental validation includes three well-known state-of-the-art MOEAs and VSD-MOEA. The MOEAs that are considered are NSGA-II [4], MOEA/D [5], and R2-EMOA [6], which can be classified as dominance-based, decomposition-based, and indicator-based, respectively. Note that for the indicator-based category, the s-metric selection evolutionary multiobjective optimisation algorithm (SMS-EMOA) (Beume et al., 2007) was also taken into account initially. However, due to its high computational cost, it is not convenient for long-term experiments as the ones used in this paper. For instance, a single execution of SMS-EMOA took more than 15 days to complete, meaning that executing several repetitions with several functions is not feasible. In the case of MOEA/D, several variants have been devised. The MOEA/D implementation considered is the one that obtained first place in the IEEE 2009 Congress on Evolutionary Computation's MOP Competition proposed by Zhang et al. (2009). The common configuration in all the experiments was as follows: the population size was set to 100, and the genetic operators were Simulated Binary Crossover (SBX) and polynomial-based mutation (Agrawal et al., 1994; Deb and Goyal, 1996). The crossover probability was set to 0.9 and the crossover distribution index was set to 2. Similarly, the mutation probability and distribution index were fixed to $1/n$ and 50, respectively. The additional parameterization required by each algorithm is shown in Table 2.

Note that scalarization functions are required in MOEA/D and R2-EMOA. In both cases, the Tchebycheff approach is used. The procedure for generating the weight vectors differs in MOEA/D and R2-EMOA. R2-EMOA was applied with 501 and 496 weight vectors for two and three objectives, respectively (Trautmann et al., 2013). In contrast, MOEA/D requires the same number of weight vectors as the population size. They were generated using the uniform design (UD) and the good lattice point (GLP) method (Ma

[4] http://jmetalcpp.sourceforge.net/
[5] http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC09-MOEA/CEC09-MOEA.htm
[6] http://inriadortmund.gforge.inria.fr/r2emoa/

Table 3: Summary of the hypervolume ratio results attained for problems with two objectives, the higher the normalized hypervolume value the better the algorithm.

	VSD-MOEA			CPDEA			MOEA/D			R2-EMOA		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	**0.993**	0.994	0.002	0.963	0.965	0.013	**0.993**	0.993	0.001	0.980	0.989	0.018
WFG2	**0.996**	0.998	0.008	0.993	0.996	0.009	0.965	0.965	0.000	0.966	0.966	0.005
WFG3	**0.992**	0.992	0.000	0.973	0.973	0.002	**0.992**	0.992	0.000	0.991	0.991	0.000
WFG4	0.990	0.990	0.000	0.964	0.964	0.003	0.988	0.988	0.000	**0.991**	0.991	0.000
WFG5	0.880	0.881	0.003	0.862	0.862	0.002	0.877	0.876	0.003	**0.882**	0.882	0.002
WFG6	0.884	0.884	0.012	0.787	0.788	0.003	**0.918**	0.919	0.020	0.914	0.914	0.015
WFG7	0.990	0.990	0.000	0.973	0.974	0.002	0.988	0.988	0.000	**0.991**	0.991	0.000
WFG8	**0.904**	0.947	0.053	0.875	0.881	0.026	0.808	0.808	0.007	0.803	0.804	0.005
WFG9	**0.946**	0.961	0.027	0.791	0.791	0.002	0.912	0.949	0.070	0.894	0.953	0.079
DTLZ1	0.992	0.992	0.000	0.991	0.991	0.000	**0.993**	0.993	0.000	0.992	0.992	0.000
DTLZ2	0.990	0.990	0.000	0.983	0.983	0.001	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ3	0.990	0.990	0.000	0.988	0.988	0.000	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ4	**0.990**	0.990	0.000	0.979	0.980	0.003	0.989	0.989	0.000	0.678	0.991	0.362
DTLZ5	0.990	0.990	0.000	0.983	0.983	0.001	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ6	**0.990**	0.990	0.000	0.807	0.820	0.088	0.989	0.989	0.000	0.685	0.667	0.088
DTLZ7	0.996	0.996	0.000	0.995	0.995	0.000	0.996	0.996	0.000	**0.997**	0.997	0.000
UF1	**0.989**	0.990	0.003	0.976	0.976	0.003	0.980	0.981	0.005	0.881	0.881	0.030
UF2	**0.987**	0.988	0.004	0.968	0.968	0.001	0.986	0.986	0.004	0.979	0.979	0.003
UF3	**0.876**	0.878	0.014	0.755	0.757	0.049	0.616	0.609	0.065	0.556	0.557	0.040
UF4	0.891	0.891	0.003	0.850	0.849	0.004	0.883	0.884	0.005	**0.900**	0.901	0.003
UF5	0.589	0.579	0.050	**0.676**	0.671	0.070	0.294	0.206	0.247	0.306	0.332	0.152
UF6	**0.854**	0.852	0.030	0.839	0.848	0.043	0.526	0.538	0.143	0.558	0.545	0.113
UF7	**0.985**	0.985	0.002	0.967	0.968	0.004	0.957	0.979	0.121	0.756	0.944	0.225
Mean	**0.943**	0.945	0.009	0.910	0.912	0.014	0.896	0.895	0.030	0.855	0.880	0.050

et al., 2014; Molinet Berenguer and Coello Coello, 2015).

Given that all the algorithms considered are stochastic, each execution was repeated 35 times with different seeds. The hypervolume indicator (HV) is used to compare results. Note that in the supplementary material, the results are also compared in terms of the IGD+ metric, with the conclusions being quite similar. The reference point used to calculate the HV is chosen to be a vector whose values are sightly larger (ten percent) than the nadir point, as suggested in Ishibuchi et al. (2017). The normalized HV is used to facilitate the interpretation of the results (Li et al., 2014), and the value reported is computed as the ratio between the normalized HV obtained and the maximum attainable normalized HV. In this way, a value equal to one means a perfect approximation. Note that a value equal to one is not attainable because MOEAs yield a discrete approximation. Finally, in order to statistically compare the HV ratios, a guideline similar to that proposed in Durillo et al. (2010) was used. First a Shapiro-Wilk test was performed to check if the values of the results followed a Gaussian distribution. If so, the Levene test was used to check for the homogeneity of the variances. If the samples had equal variance, an ANOVA test was done; if not, a Welch test was performed. For non-Gaussian distributions, the non-parametric Kruskal-Wallis test was used to test whether samples are drawn from the same distribution. An algorithm X is said to beat algorithm Y when the differences between them are statistically significant, and the mean and median HV ratios obtained by X are higher than the mean and median achieved by Y.

4.1 Comparison against State-of-the-art MOEAs in long-term executions

Our first experiment aims to compare the long-term performance of VSD-MOEA against state-of-the-art proposals, which is the kind of execution where diversity-based EAs have been more successful. Specifically, the stopping criterion was set to 2.5×10^7 function evaluations.

Table 3 shows the HV ratio obtained for the benchmark functions with two objectives. Specifically, the minimum, maximum, mean and standard deviation of the

Table 4: Statistical Tests and Deterioration Level of the HV ratio for problems with two objectives

	↑	↓	↔	Score	Deterioration
MOEA/D	24	36	9	-12	1.615
NSGA-II	13	49	7	-36	1.496
R2-EMOA	34	21	14	13	1.597
VSD-MOEA	50	15	4	35	0.059

HV ratio is shown for each method and problem tested. The last row shows the results considering all the test problems together. For each test problem, the data for the method that yielded the largest mean is shown in **boldface**. Additionally, all the methods that were not statistically inferior than the method with the largest mean are shown in **boldface**. From here on, the methods shown in **boldface** for a given problem are referred to as the winning methods. Based on the number of test problems where each method is in the group of the winning methods for the cases with two objectives, the best methods are VSD-MOEA and R2-EMOA with 12 and 8 wins, respectively. Thus, VSD-MOEA is the most competitive method in terms of this metric. More impressive is the fact that the mean HV ratio attained by VSD-MOEA, when all the problems are considered simultaneously, is much higher than the one attained by R2-EMOA. In fact, the total means of R2-EMOA (0.882), NSGA-II (0.886) and MOEA/D (0.881) are quite similar. In contrast, VSD-MOEA achieved a much higher value (0.949). Inspecting the data carefully, it is clear that in the cases where VSD-MOEA loses, the difference with respect to the best method is not very large. For instance, the difference between the mean HV ratio attained by the best method and by VSD-MOEA was never larger than 0.1. However, all the other methods exhibited a deterioration greater than 0.1 in several cases. Specifically, it happened in 5, 5 and 6 problems for R2-EMOA, NSGA-II and MOEA/D, respectively. This means that even if VSD-MOEA loses in some cases, its deterioration is always small, exhibiting a much more robust behavior than any other method.

In order to better clarify these findings, pair-wise statistical tests were done among each method tested in each test problem. For the two-objective cases, Table 4 shows the number of times that each method won (column ↑), lost (column ↓), tied (column ↔) and **Score**. The later is calculated as the diference between the number of times that each method won and the numer of times that each method lost. Additionally, for each method M, we calculated the sum of the differences between the mean HV ratio attained by the best method (the ones with the highest mean) and method M, for each problem where M was not in the group of winning methods. This value is shown in the Deterioration column. The data confirms that although VSD-MOEA loses in some cases, the overall numbers of wins and losses favors VSD-MOEA. More importantly, the total deterioration is quite lower in the case of VSD-MOEA, confirming that when VSD-MOEA loses, the deterioration is not that large.

Tables 5 and 6 show the same information for the problems with three objectives. In this case, the superiority of VSD-MOEA is even clearer. Taking into account the mean of all the test problems, VSD-MOEA again obtained a much larger mean HV ratio than the other methods. Specifically, VSD-MOEA obtained a value of 0.918, whereas the second ranked algorithm (R2-EMOA) obtained a value of 0.855. Once again, the difference between the mean HV ratio obtained by the best method and by VSD-MOEA was never greater than 0.1. However, all the other methods exhibited a deterioration greater than 0.1 in several cases. In particular, this happened in 5, 6 and 6 problems for R2-EMOA, NSGA-II and MOEA/D, respectively. Moreover, in this case, VSD-MOEA is much supe-

Table 5: Summary of the hypervolume ratio results attained for problems with three objectives, the higher the normalized hypervolume value the better the algorithm.

	VSD-MOEA			CPDEA			MOEA/D			R2-EMOA		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	0.785	0.789	0.017	0.430	0.428	0.033	**0.968**	0.968	0.001	0.928	0.926	0.009
WFG2	**0.988**	0.989	0.001	0.923	0.924	0.006	0.967	0.976	0.034	0.905	0.962	0.070
WFG3	0.989	0.989	0.000	0.906	0.905	0.013	**0.993**	0.992	0.000	0.992	0.992	0.000
WFG4	**0.920**	0.920	0.001	0.795	0.796	0.013	0.861	0.861	0.003	0.906	0.905	0.001
WFG5	0.834	0.832	0.004	0.801	0.801	0.004	0.795	0.795	0.001	**0.842**	0.843	0.002
WFG6	0.837	0.835	0.007	0.766	0.767	0.005	0.811	0.810	0.012	**0.860**	0.860	0.007
WFG7	**0.919**	0.919	0.001	0.774	0.778	0.021	0.865	0.865	0.000	0.905	0.905	0.001
WFG8	**0.863**	0.864	0.035	0.672	0.682	0.039	0.779	0.779	0.002	0.820	0.820	0.002
WFG9	**0.822**	0.824	0.038	0.727	0.727	0.005	0.810	0.837	0.047	0.804	0.772	0.048
DTLZ1	**0.965**	0.965	0.001	0.964	0.964	0.001	0.950	0.950	0.000	0.940	0.940	0.001
DTLZ2	**0.930**	0.930	0.001	0.864	0.864	0.017	0.899	0.899	0.000	0.915	0.915	0.001
DTLZ3	**0.930**	0.930	0.001	0.830	0.916	0.239	0.899	0.899	0.000	0.912	0.915	0.004
DTLZ4	**0.930**	0.930	0.001	0.859	0.858	0.006	0.899	0.899	0.000	0.652	0.577	0.257
DTLZ5	**0.986**	0.986	0.000	0.977	0.977	0.002	0.978	0.978	0.000	**0.986**	0.986	0.000
DTLZ6	**0.986**	0.986	0.000	0.660	0.643	0.115	0.978	0.978	0.000	0.775	0.760	0.082
DTLZ7	**0.965**	0.965	0.001	0.940	0.941	0.004	0.914	0.914	0.000	0.852	0.852	0.014
UF8	**0.918**	0.920	0.011	0.699	0.711	0.045	0.778	0.777	0.006	0.853	0.905	0.104
UF9	**0.962**	0.965	0.011	0.784	0.793	0.053	0.792	0.747	0.071	0.844	0.783	0.076
UF10	**0.602**	0.581	0.095	0.122	0.121	0.060	0.309	0.270	0.150	0.268	0.209	0.132
Mean	**0.902**	0.901	0.012	0.763	0.768	0.036	0.855	0.852	0.017	0.840	0.833	0.043

Table 6: Statistical Tests and Deterioration Level of the HV ratio for problems with three objectives

	↑	↓	↔	Score	Deterioration
MOEA/D	16	37	4	-21	1.601
NSGA-II	9	45	3	-36	2.557
R2-EMOA	31	22	4	9	1.223
VSD-MOEA	52	4	1	47	0.037

rior than the other methods not only in terms of total deterioration, but also in terms of total wins and losses (see Table 6 and data shown in **boldface** in Table 5). VSD-MOEA was in the group of winning methods for 16 out of 19 test problems, whereas the second best-ranked algorithm (R2-EMOA) was in the group of winning methods for only 3 test problems.

4.2 Decision Variable Scalability Analysis

In order to study the scalability of VSD-MOEA in terms of the number of decision variables, all of the algorithms already described were tested with the same benchmark problems, but considering 50, 100, and 250 variables. Note that in the WFG test problems, the number of position parameters (k) and distance parameters (l) must be specified. Specifically, the number of distance parameters was set to 42, 84, and 210 when using 50, 100 and 250 variables, respectively. The rest of the decision variables were position parameters, meaning that they were 8, 16 and 40, respectively. Note that increasing the number of variables greatly increases the computing time required. As a result, this study takes into account middle-term executions. Specifically, the stopping criterion was set to 2.5×10^6 function evaluations. Figure 2 shows the mean HV ratio for the four algorithms tested, considering the problems with two and three objectives. As expected, the HV ratio decreases as the number of variables increases. In the two-objective case, the deterioration is similar in every algorithm, so the superiority of VSD-MOEA is clear regardless of the number of variables. In contrast, in the three-objective case, the deterioration of VSD-MOEA is higher than for R2-EMOA and MOEA/D. In fact, when considering 250 variables, the performance of VSD-MOEA is just slightly superior to that of R2-EMOA.

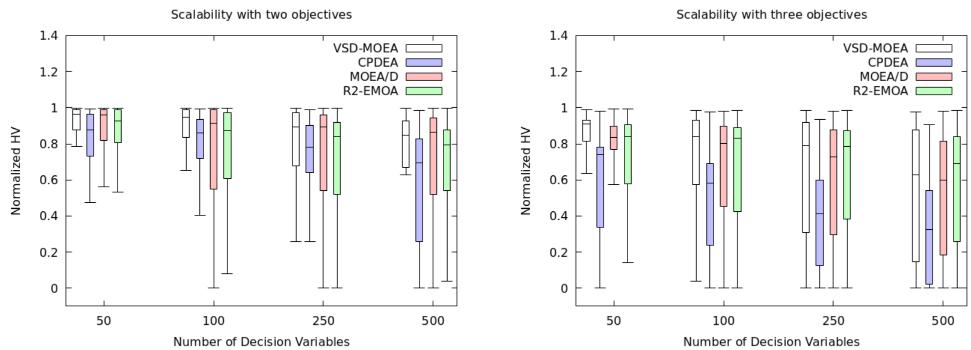

Figure 2: Mean of the HV ratio for 35 runs for the two-objective (left side) and three-objective (right side) problems considering different numbers of variables

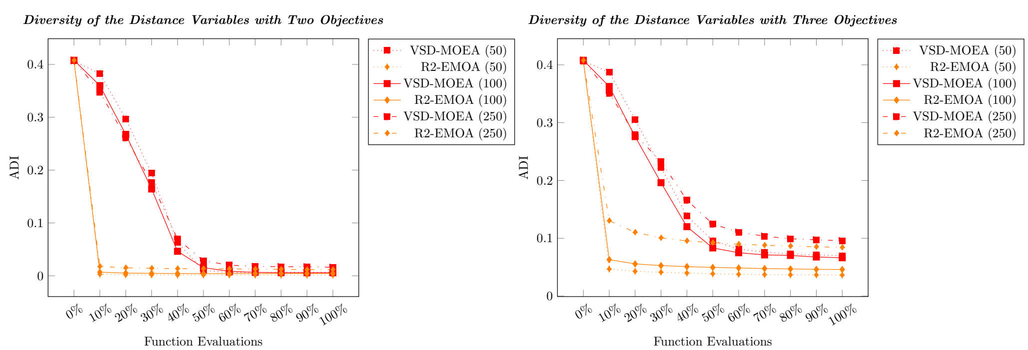

Figure 3: Evolution of ADI for problems WFG1-WFG7 with two and three objectives considering only the distance variables

In order to better understand this behavior, we selected problems WFG1 to WFG7. The WFG test problems divide the variables into two kinds of parameters (this framework uses the term parameter instead of variable): the distance parameters and the position parameters. Note that a parameter i is a distance parameter when for all $\vec{\mathbf{x}}$, modifying x_i results in a new solution that dominates $\vec{\mathbf{x}}$, is equivalent to $\vec{\mathbf{x}}$, or is dominated by $\vec{\mathbf{x}}$. However, if i is a position parameter, modifying x_i in $\vec{\mathbf{x}}$ always results in a vector that is incomparable or equivalent to $\vec{\mathbf{x}}$ (Huband et al., 2005b). Additionally, note that we selected problems WFG1-WFG7 because their distance parameter values associated to all Pareto optimal solutions have exactly the same values:

$$x_{i=k+1:n} = 2i \times 0.35 \tag{5}$$

This is very important because it has been shown that for these cases, state-of-the-art MOEAs might provoke a quick convergence in *distance parameters*, resulting in an effect that is similar to premature convergence in the single-objective case (Kukkonen and Lampinen, 2009; Castillo et al., 2017).

For each algorithm, we calculated the average (mean) Euclidean distance among individuals (ADI) in the population by considering only the distance parameters. Figure 3 shows how the ADI evolves for the two-objective (left side) and three-objective (right side) problems. The behavior of NSGA-II and MOEA/D — which are not included — is similar to that of R2-EMOA in terms of how the ADI evolves. Thus, to avoid saturating these figures, only the information for VSD-MOEA and R2-EMOA with 50, 100 and 250 variables is shown. The first obvious fact is that VSD-MOEA converges much slower than R2-EMOA. Accordingly, the difference between the diversity maintained in the first generation and that maintained after 10% of the execution, is much larger in R2-EMOA than in VSD-MOEA. In the case of VSD-MOEA, the decrease in ADI is quite linear until the halfway point of the execution. This is due to the way in which the threshold distance value (D_t) is calculated. Additionally, a closer inspection of the data reveals other important aspects that must be discussed. In the two-objective case, increasing the number of variables causes the diversity in the R2-EMOA to increase slightly. However, the amount of diversity is low even when using 250 variables, meaning that incorporating mechanisms to increase diversity — as is done in VSD-MOEA — is very helpful. In contrast, in the three-objective case, the amount of diversity in R2-EMOA is not as low. Moreover, increasing the number of variables yields a significant increase in the resulting ADI, meaning that in this case, fast convergence is not an important issue. These results show that, as the number of objectives and variables increases, MOEAs tend to maintain a higher variable space diversity in an implicit way, meaning that explicitly controlling the variable space diversity is probably not as important.

Finally, it is worth noting that we selected some problems to conduct long-term executions with 250 variables. VSD-MOEA was able to further improve the results when using long-term executions, while the other state-of-the-art algorithms did not yield significant improvements. This probably means that as technology evolves, allowing longer executions to be carried out in reasonable time frames, the incorporation of explicit control of diversity will be even more important. Note that this also happens in the single-objective case, where the benefits of explicitly controlling diversity appears only when using executions lasting several weeks when dealing with large instances of the Traveling Salesman Problem (Segura et al., 2015).

4.3 Analysis of the Stopping criterion

This section ilustrates the main reason behind the superiority of the VSD-MOEA against state-of-the-art algorithms in long-term executions. As previously discussed, EAs with explicit control of diversity often are more useful in long-term executions. The fact that we selected a rather large stopping criterion in our first experiment might lead readers to think that VSD-MOEA is only useful in extremely long-term executions; however, this is not the case. In this section we analyze the performance of VSD-MOEA and state-of-the-art algorithms with several stopping criteria, i.e., maximum number of function evaluations. Three different ranges were explored for the stopping criterion. Each range was split into ten equally distributed intervals, and experiments were run with each different number of function evaluations. The ranges considered were $[2.5{\times}10^4, 2.5{\times}10^5]$, $[2.5{\times}10^5, 2.5{\times}10^6]$ and $[2.5{\times}10^6, 2.5{\times}10^7]$. These ranges are referred to as short-term, middle-term and long-term executions, respectively. Note that state-of-the-art algorithms can be executed just once (with 2.5×10^7 function evaluations) by saving the intermediate results. However, VSD-MOEA makes decisions that depend on the stopping criteria, so independent executions were required for each stopping criterion.

Figure 4 shows the mean HV ratio obtained with each MOEA with two and three objectives, respectively. All the problems were considered to calculate this mean ratio. Each figure is divided into three graphs corresponding to short-term, middle-term

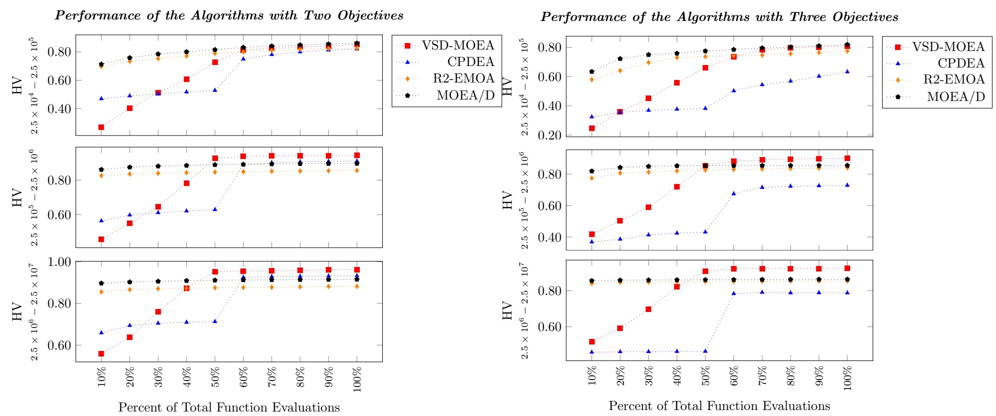

Figure 4: Performance of MOEAs for the problems with two objectives (left side) and three objectives (right side) considering three ranges for the stopping criterion: short-term (first row), middle-term (second row) and long-term (third row).

and long-term. In the two-objective case, for the shortest executions, VSD-MOEA is not very competitive. In the range $[2.5 \times 10^4, 7.5 \times 10^4]$, it exhibits the worst performance, meaning that for very short-term executions, explicitly promoting additional diversity is not helpful. When using 10^5 function evaluations, the resulting HV ratio is similar than that obtained by other methods. Finally, when using more than 2.5×10^5 function evaluations, the HV ratio obtained by VSD-MOEA is much higher than the one obtained by other methods. It is worth noting that VSD-MOEA is the only method that truly takes advantage of using long-term executions, with the remaining methods just showing a slight improvement. In the three-objective case, VSD-MOEA yields a lower HV ratio than R2-EMOA and MOEA/D in short-term executions, but as more function evaluations are executed, the differences decrease. In this case, after 5×10^5 function evaluations, the performance of VSD-MOEA is similar to that of R2-EMOA. Finally, as in the two-objective case, with additional function evaluations, the differences between VSD-MOEA and the remaining algorithms increase in favor of VSD-MOEA. Thus, while the most important benefits arise in long-term executions, users can benefit from VSD-MOEA even in shorter executions.

4.4 Analysis of the Initial Threshold Value

One of the disadvantages of including a strategy for controlling diversity is that this is usually done at the expense of incorporating additional parameters in the EA designed. In the case of VSD-MOEA, the initial threshold value (D_I) must be set. The higher this value is, the greater the exploration of the decision variable space. Note that in all the previous experiments, $D_I = 0.4$ was used. This value was selected based on some preliminary experiments. This section is devoted to analyzing the performance of VSD-MOEA when using different D_I values. Note that, since normalized distances are used, the maximum difference that can appear is 1. Additionally, note that when D_I is set to 0, no individual is penalized on the basis of its decision variable space diversity contribution, so VSD-MOEA would behave like a more traditional MOEA. As a result, the values $D_I = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ were tested. As in previous experiments, the whole set of benchmark problems was used and the stopping criterion was set to 2.5×10^7 function evaluations.

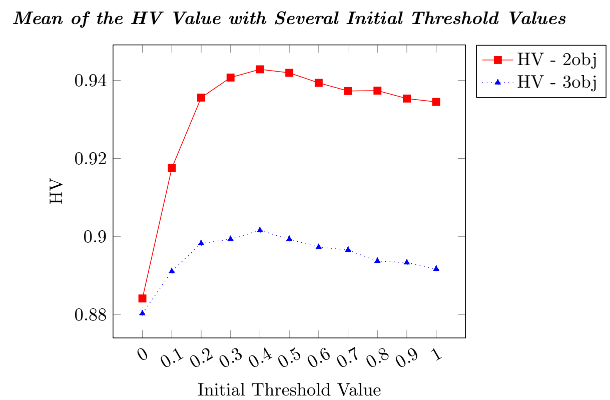

Figure 5: Mean of HV values taking into account all the problems with several initial threshold values

Figure 5 shows the mean HV ratio obtained for both the two-objective and the three-objective case. Note that even when D_I is set to 0, VSD-MOEA yielded better HV ratios than other state-of-the-art algorithms (see Tables 3 and 5). Specifically, the values were 0.912 and 0.893 for two and three objectives, respectively. This means that the novel density estimator put forth in this paper is indeed helpful. However, the increase in performance when using other D_I values is clear. The HV ratio obtained quickly increases as higher D_I values up to 0.4 are used. Then, with values in the range $[0.5, 0.9]$, the performance decreases slightly. There is a large range of values where the performance is very good, meaning that the behavior of VSD-MOEA is quite robust. Thus, properly setting this parameter is not a complex task.

5 Conclusions and Future Work

EAs have been one of the most popular approaches for dealing with complex optimization problems. Their design is a highly complex task that requires defining several components. Looking at the differences between single-objective and multi-objective optimizers, it is worth noting that several state-of-the-art single-objective optimizers explicitly consider the diversity of the variable space, particularly when dealing with long-term executions, whereas this is not the case for MOEAs.

This paper proposes a novel MOEA, called VSD-MOEA, that takes into account the diversity of both decision variable space and objective function space. The main novelty is that the importance given to the different diversities is adapted during the optimization process. In particular, in VSD-MOEA more importance is given to the diversity of decision variable space at the initial stages, but as evolution progresses it gradually grants more importance to the diversity of objective function space. This is performed using a penalty method that is integrated into the replacement phase. Also included is a novel density estimator based on IGD+ that is used to select from the non-penalized individuals.

The experimental validation carried out shows a remarkable improvement in VSD-MOEA when compared to state-of-the-art MOEAs both in two-objective and three-objective problems. Moreover, our proposal not only improves the state-of-the-art algorithms in long-term and medium-term executions, but it also offers a competitive performance in short-term executions. The scalability analyses show that as the number of objectives and decision variables increases, the implicit variable space maintained by state-of-the-art MOEAs also increases. Thus, for large number of objectives and decision variables, explicitly considering the diversity of decision variable space is less helpful. Finally, the analysis of the initial threshold distance, which is an additional parameter required by VSD-MOEA, shows that finding a proper value for this parameter is not a difficult task.

In the future, we plan to apply the principles studied in this paper to other categories of MOEAs. For instance, including the diversity management put forth in this paper in decomposition-based and indicator-based MOEAs seems plausible. Additionally, we would like to develop an adaptive scheme to avoid setting the initial threshold value. Finally, in order to obtain even better results, these strategies are going to be incorporated into a multi-objective memetic algorithm.

Acknowledgments

The first two authors acknowledge the financial support from CONACyT through the "Ciencia Básica" project no. 285599. The last author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a project from the 2018 SEP-Cinvestav Fund (application no. 4).

References

Agrawal, R. B., Deb, K., Deb, K., and Agrawal, R. B. (1994). Simulated binary crossover for continuous search space. Technical report.

Bechikh, S., Datta, R., and Gupta, A. (2016). *Recent advances in evolutionary multi-objective optimization*, volume 20. Springer.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume . *European Journal of Operational Research*, 181(3):1653 – 1669.

Castillo, J. C., Segura, C., Aguirre, A. H., Miranda, G., and León, C. (2017). A multi-objective decomposition-based evolutionary algorithm with enhanced variable space diversity control. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1565–1571.

Chan, K. and Ray, T. (2005). An evolutionary algorithm to maintain diversity in the parametric and the objective space. In *International Conference on Computational Robotics and Autonomous Systems (CIRAS), Centre for Intelligent Control, National University of Singapore*.

Chiang, T. C. and Lai, Y. P. (2011). MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1473–1480.

Deb, K. and Abouhawwash, M. (2016). An optimality theory-based proximity measure for set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(4):515–528.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45.

Deb, K. and Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145.

Deb, K. and Tiwari, S. (2005). Omni-optimizer: A procedure for single and multi-objective optimization. In *EMO*, volume 3410, pages 47–61. Springer.

Durillo, J. J., Nebro, A. J., Coello, C. A. C., Garcia-Nieto, J., Luna, F., and Alba, E. (2010). A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems. *IEEE Transactions on Evolutionary Computation*, 14(4):618–635.

Herrera, F. and Lozano, M. (2003). Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft Computing*, 7(8):545–562.

Herrera, F., Lozano, M., et al. (1996). Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, 8(1996):95–125.

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1.

Huband, S., Barone, L., While, L., and Hingston, P. (2005a). *A Scalable Multi-objective Test Problem Toolkit*, pages 280–295. Springer Berlin Heidelberg, Berlin, Heidelberg.

Huband, S., Barone, L., While, L., and Hingston, P. (2005b). A scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer.

Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2017). Reference point specification in hypervolume calculation for fair comparison and efficient search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 585–592. ACM.

Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. (2015). *Modified Distance Calculation in Generational Distance and Inverted Generational Distance*, pages 110–125. Springer International Publishing, Cham.

Ishibuchi, H. and Shibata, Y. (2003). *An Empirical Study on the Effect of Mating Restriction on the Search Ability of EMO Algorithms*, pages 433–447. Springer Berlin Heidelberg, Berlin, Heidelberg.

Koumousis, V. and Katsaras, C. (2006). A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. 10(1):19–28.

Kukkonen, S. and Lampinen, J. (2009). Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In *2009 IEEE Congress on Evolutionary Computation*, pages 1943–1950. IEEE.

Li, K., Deb, K., Zhang, Q., and Kwong, S. (2014). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

Li, X., Epitropakis, M. G., Deb, K., and Engelbrecht, A. (2016). Seeking multiple solutions: an updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*, 21(4):518–538.

Liang, J., Yue, C., and Qu, B. (2016). Multimodal multi-objective optimization: A preliminary study. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2454–2461. IEEE.

Ma, X., Qi, Y., Li, L., Liu, F., Jiao, L., and Wu, J. (2014). Moea/d with uniform decomposition measurement for many-objective problems. *Soft Computing*, 18(12):2541–2564.

Mengshoel, O. J., Galán, S. F., and de Dios, A. (2014). Adaptive generalized crowding for genetic algorithms. *Information Sciences*, 258:140 – 159.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

Molinet Berenguer, J. A. and Coello Coello, C. A. (2015). *Evolutionary Many-Objective Optimization Based on Kuhn-Munkres Algorithm*, pages 3–17. Springer International Publishing, Cham.

Pandey, H. M., Chaudhary, A., and Mehrotra, D. (2014). A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing*, 24:1047 – 1077.

Rudolph, G., Naujoks, B., and Preuss, M. (2007). Capabilities of emoa to detect and preserve equivalent pareto subsets. In *Evolutionary Multi-Criterion Optimization*, pages 36–50. Springer.

Segura, C., Botello Rionda, S., Hernández Aguirre, A., and Valdez Peña, S. I. (2015). A novel diversity-based evolutionary algorithm for the traveling salesman problem. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 489–496. ACM.

Segura, C., Coello, C. A. C., Segredo, E., and Aguirre, A. H. (2016). A Novel Diversity-Based Replacement Strategy for Evolutionary Algorithms. *IEEE Transactions on Cybernetics*, 46(12):3233–3246.

Segura, C., Hernández-Aguirre, A., Luna, F., and Alba, E. (2017). Improving diversity in evolutionary algorithms: New best solutions for frequency assignment. *IEEE Transactions on Evolutionary Computation*, 21(4):539–553.

Shir, O. M., Preuss, M., Naujoks, B., and Emmerich, M. T. (2009). Enhancing decision space diversity in evolutionary multiobjective algorithms. In *EMO*, volume 9, pages 95–109. Springer.

Tan, K. C., Khor, E. F., and Lee, T. H. (2005). *Multiobjective Evolutionary Algorithms and Applications (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Toffolo, A. and Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary computation*, 11(2):151–167.

Trautmann, H., Wagner, T., and Brockhoff, D. (2013). R2-emoa: Focused multiobjective search using r2-indicator-based selection. In *International Conference on Learning and Intelligent Optimization*, pages 70–74. Springer.

Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2016). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.

Ulrich, T., Bader, J., and Zitzler, E. (2010). Integrating decision space diversity into hypervolume-based multiobjective search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 455–462. ACM.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and Exploitation in Evolutionary Algorithms: A Survey. *ACM Computing Surveys*, 45(3):35:1–35:33.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475 – 489.

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

Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *2009 IEEE congress on evolutionary computation*, pages 203–208. IEEE.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, 264.

Zhou, A., Zhang, Q., and Jin, Y. (2009). Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE transactions on evolutionary computation*, 13(5):1167–1189.

Zitzler, E. and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer.