An enhanced adaptive geometry evolutionary algorithm using stochastic diversity mechanism

Fodil Benali, Damien Bodénès, Cyril de Runz, Nicolas Labroche

An Enhanced Adaptive Geometry Evolutionary Algorithm Using Stochastic Diversity Mechanism

Fodil Benali*
Damien Bodénès
fbenali@adwanted.com
dbodenes@adwanted.com
Adwanted Group
Paris, France

Cyril De Runz
Nicolas Labroche
cyril.derunz@univ-tours.fr
nicolas.labroche@univ-tours.fr
BDTLN - LIFAT, University of Tours
Blois, France

ABSTRACT

Evolutionary Algorithms have been regularly used for solving multi and many objectives optimization problems. The effectiveness of such methods is determined generally by their ability to generate a well-distributed front (diversity) that is as close as possible to the optimal Pareto front (proximity). Analysis of current multi-objective evolutionary frameworks shows that they are still sub-optimal and present poor versatility on different geometries and dimensionalities. For that, in this paper, we present AGE-MOEA++, a new Multi and Many Objective Evolutionary Algorithm that: (1) incorporates the principle of Pareto Front (PF) shape fitting to enhance the convergence in different shaped high dimensional objective spaces, and (2) adapts K-means ++ fundamentals in order to best manage the diversity in non-uniform distributed PF. The empirical study shows that our proposal has better results than the state-of-the-art approaches in terms of IGD and is competitive in terms of GD.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**.

KEYWORDS

K-Means++ initialization, Non-Euclidean-Geometry, Evolutionary Computation

ACM Reference Format:
Fodil Benali, Damien Bodénès, Cyril De Runz, and Nicolas Labroche. 2022. An Enhanced Adaptive Geometry Evolutionary Algorithm Using Stochastic Diversity Mechanism. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3512290.3528820

1 INTRODUCTION

A multi-objective optimization (MOP) problem with $\mathcal{M}$ objectives to be minimized can be written as follows [10]:

$$Minimize\ f(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_{\mathcal{M}}(\vec{x})]^T, \quad (1)$$

$$Subject\ to\ \vec{x} \in\ \mathbb{U}\ \subset \mathbb{R}^n \quad (2)$$

where $f_i(x)$ is the i^{th} objective function ($i \in [\![1; \mathcal{M}]\!]$), $\vec{x}$ is an n-dimensional decision vector, and $\mathbb{U}$ is the feasible region of $\vec{x}$. The objective function vector f maps a solution $\vec{x}$ in an n-dimensional decision space to a point $([f_1(\vec{x}), f_2(\vec{x}), \ldots, f_{\mathcal{M}}(\vec{x})]^T)$ in an $\mathcal{M}$-dimensional objective space. A multi-objective problem with $\mathcal{M} > 3$ is often referred to as many-objective problem.

In general, a MOP does not have one single optimal solution since objectives are conflicting with each other. Therefore, no single solution simultaneously optimizes all objectives. Thus, solutions are compared using the Pareto dominance principle. A solution $\vec{x}$ dominates another solution $\vec{y}$ (denoted by $\vec{x} \succ \vec{y}$), if $f_i(\vec{x}) \leq f_i(\vec{y})\ \ \forall i \in [\![1; \mathcal{M}]\!]$ and there exists at least one index $j \in [\![1; \mathcal{M}]\!]$ such that $f_j(\vec{x}) < f_j(\vec{y})$. If a solution $\vec{x}^*$ is not dominated by any other solutions, $\vec{x}^*$ is said to be Pareto optimal solution. The set of all Pareto optimal solutions is the Pareto optimal solution set (PS). The projection of PS in the objective space is called the Pareto front (PF) or Pareto frontier.

A large number of evolutionary algorithms (EAs) have been proposed in the literature [4, 5, 13, 14, 18, 22, 24, 27–29] to solve Multi and Many Objective optimization problems. The goal of each algorithm is to approximate the optimal Pareto front with a set of non-dominated solutions that are as close as possible to PF (**proximity**), and that is well-distributed over the optimal PF (**diversity**).

Multi Objective Evolutionary Algorithms (MOEAs) are often categorized into three classes based on their fitness evaluation mechanisms [20]. First, decomposition-based algorithms such as MOEA/D [28], NSGA- III [13] and [18, 27], where a MOP is decomposed into a number of single-objective problems. Each single-objective problem has the same scalarizing function and a different weight vector. A single solution is assigned to each single-objective problem. All single- objective problems are optimized in a cooperative manner towards different directions in the objective space along the weight vectors. However such methods rely on multiple parameters (e.g., setup of the reference lines) which is not adequate for a decision maker in an industrial context.

Second, Pareto dominance-based algorithms such as SPEA [29] and NSGA-II [14], where the Pareto dominance relation among solutions is used as the primary fitness evaluation criterion and a crowding distance to promote diversity. However, it was reported in many studies [12, 21, 23] that the performance of Pareto dominance-based severely degrades with the increase in the number of objectives.

*Also with BDTLN - LIFAT, University of Tours.

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528820

This is because almost all solutions become non-dominated with each other in very early generations. As consequence, solutions are not comparable with each other, the same fitness value is assigned to all solutions by the Pareto dominance criterion. Thus, no strong selection pressure is generated to push the population towards the PF. Finally, Indicator-based algorithms, such as HV-based methods [4], work well on many-objective problems. However, they have a large computation load for calculating the HV contribution of each solution.

Another problem with a number of MOEAs mentioned above is that they are built upon the implicit assumption that the PF (or its approximation) has an Euclidean geometry. For example, in NSGA-III the reference points are generated using Das and Dennis's systematic approach [9], which places points on a flat hyper-surface. However, in many MOPs, the PF is convex (i.e., hyperbolic geometry) or concave (i.e., spherical geometry). As consequence, this impacts greatly the convergence and diversity of the obtained solutions.

To overcome such limitations, a new MOEA has been proposed recently, called AGE-MOEA [22] (Adaptive Geometry Estimation based MOEA), for evolutionary multi and many-objective optimization. The idea consists in fitting the geometry of the PF in order to adapt diversity and convergence mechanisms on the Pareto Front shape. AGE-MOA is the best in terms of convergence and diversity of solutions on the Pareto Front. However, it has a very time consuming mechanism to promote diversity (a high computational complexity $O(MN^2 + N^3)$). This is due to the use of deterministic-based diversity promoting mechanism that penalizes the performance of the algorithm greatly.

The problem of data selection in a non-uniform distribution in order to guarantee certain diversity has been widely discussed in several data mining tasks. In particular, in clustering methods like k-means, where it has been demonstrated that the quality of clustering is extremely sensitive to initial centroids selection. To remedy to this problem, there has been an extremely efficient and simple algorithm for selecting data in a non-uniform distribution called k-mean++ [2]. Augmenting k-means with this stochastic approach (k-means++) has shown that the obtained initial set of centers is provably competitive with the optimal solution. Consequently, the obtained performance is O(log k)-competitive with the optimal clustering.

For that, in this work, we propose AGE-MOEA++, a new Multi and Many Objective Evolutionary Algorithm that: (1) incorporates the principle of PF shape Fitting to enhance the convergence in different shaped high dimensional objective spaces, and (2) adapts K-means ++ fundamentals in order to best manage the diversity in non-uniform distributed PF. Our contributions are the following: (i) we formalize the problem of diversity maximization in MOP, (ii) we propose a simple, parameter-less, Multi and Many Objective Evolutionary Algorithm with a fast and simple stochastic diversity mechanism ($O(MN^2)$[3]), and (iii) we describe experiments that show to which extent the proposal is able to balance between convergence and diversity in different shaped and dimensional test problems.

The remainder of the paper is organized as follows: Section 2 describes $\mathcal{L}_p$ norms and their relation with curvatures and shapes of the hyper-surfaces. Section 3 details the proposed framework AGE-MOEA++ while Section 4 describes the empirical study and results. Finally, Section 5 concludes and opens future work perspectives.

2 PRELIMINARIES: $\mathcal{L}_p$ NORMS AND NON-EUCLIDEAN GEOMETRY

A norm on a real linear space $\mathcal{X}$ is a mapping $\|.\|$ from $\mathcal{X}$ into $\mathbb{R}^+$ that satisfy the following axioms:

(i) $\|\vec{x}\| \geq 0$ with equality if and only if $x = 0$,
(ii) $\|\alpha\vec{x}\| = |\alpha|.\|\vec{x}\|$ $(\alpha \in \mathbb{R})$,
(iii) $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$

In an M-dimensional Euclidean space $\mathbb{R}^M$, the norm (the length) of a vector $\vec{v} = (v_1, \ldots, v_M)$ is defined by:

$$\|\vec{v}\|_2 = (v_1^2 + v_2^2 + \cdots + v_M^2)^{1/2} \tag{3}$$

In the Euclidean geometry, the distance between two points A and B is the length (norm) of the straight line segment connecting the two points, i.e., $d(A, B) = \|A - B\|_2$. However, it is a well known fact that the higher the dimensionality of a space, the more sparse the space is. Here, one of the concerns is that the *Euclidean distance* loses its ability to depict spacing in high dimensions. As a consequence, the Euclidean norm does not necessarily provide the most accurate measure of the distance between two points in a high dimensional space [25]. A generalization of the Euclidean norm is the $\mathcal{L}_p$ norm, given by:

$$\|\vec{v}\|_p = (v_1^p + v_2^p + \cdots + v_M^p)^{1/p} \tag{4}$$

In fact, the set of points equidistant from a reference point (e.g., the origin of the axes) varies depending on the used $\mathcal{L}_p$ norm. The set of all points with a distance $\|.\|_p = 1$ to the origin of the axes forms a unit hyper-surface. The curvature of the hyper-surface strictly depends on the value of the exponent p. To better understand this aspect, let us consider a bi-dimensional space. For $p = 1$, the unit curve is flat and corresponds to the straight line connecting the point (0,1) and (1,0). Therefore, all points that are equidistant to the origin of the axes lie on that straight line. For $p > 1$, the unit curve is concave and when $p = 2$ it corresponds to the unit circle, i.e., a circle with a radius equal to one. Instead, for $p < 1$, the equidistant points lie on a hyperbolic curve. Therefore, the value of p determines the curvature of the unit hyper-surface associated with the norm.

In this work, we propose to adapt the norm used to evaluate the distance between solutions and the Pareto front to converge more efficiently towards optimal solutions.

3 PROPOSED ALGORITHM

With the aim of developing a tool that should be able to (i) converge to solutions optimizing a large number of objectives simultaneously, (ii) assist the Decision Maker (DM) in choosing the most suitable configurations by providing a diverse set of solutions, and (iii) be fast in order to obtain efficient solutions within a limited computational time budget.

The proposed framework, called AGE-MOEA++, inherits from the computational fast and elitist framework NSGA-II, as illustrated in Algorithm 1. It incorporates K-means++ initialization heuristic in order to serve the DM with a well distributed and diverse set of

solutions and uses the AGE-MOEA Pareto Front geometry estimation technique to be adapted to different shapes of the Pareto Front. The objective is to have an efficient management for both of the diversity and convergence towards the Pareto Optimal Front.

Algorithm 1: AGE-MOEA++ Framework.

Input: Number of Objectives $\mathcal{M}$, Size of the Population $\mathcal{N}$;
Output: Final population $\mathcal{P}$;
1 **begin**
2 $\mathcal{P} \longleftarrow$ Generate-Initial-Population($\mathcal{N}$);
3 **while** *not (Stopping-Criterion)* **do**
4 $\mathcal{Q} \longleftarrow$ Generate-Offspring($\mathcal{P}$);
5 $\mathcal{P} \longleftarrow \mathcal{P} \cup \mathcal{Q}$;
6 $\mathcal{F} \longleftarrow$ Evaluate($\mathcal{P}$);// with $\mathcal{M}$ objectives
7 $\mathcal{P} \longleftarrow$ Select-Survival($\mathcal{P}$, $\mathcal{F}$, $\mathcal{N}$);
8 **end**
9 **return** $\mathcal{P}$;
10 **end**

A general description of the proposed framework is presented in Section 3.1.

3.1 Overview

Algorithm 2: AGE-MOEA++ Selection Mecanism.

Input: Population $\mathcal{P}$, Target population size $\mathcal{N}$;
Output: Next generation population $\mathcal{P}_{new}$;
1 **begin**
2 $\mathbb{F} \longleftarrow$ Fast-Non-Dominated-Sort($\mathcal{P}$);
3 $\mathbb{F} \longleftarrow$ Normalize($\mathbb{F}$);
4 $p \longleftarrow$ Fit-P-Norm($\mathbb{F}_1$);
5 $\mathcal{P}_{new} \longleftarrow \emptyset$;
6 Calculate-Survival-Score($\mathbb{F}_1$, p, 1);
7 $r \longleftarrow 1$;
8 **while** $|\mathcal{P}_{new}| + |\mathbb{F}_r| \leq \mathcal{N}$ **do**
9 $\mathcal{P}_{new} \longleftarrow \mathcal{P}_{new} \cup \mathbb{F}_r$;
10 $r \longleftarrow r + 1$;
11 Calculate-Survival-Score($\mathbb{F}_r$, p, r);
12 **end**
13 Sort($\mathbb{F}_r$);
14 $\mathcal{P}_{new} \longleftarrow \mathcal{P}_{new} \cup \mathbb{F}_r[1 : (\mathcal{N} - |\mathcal{P}_{new}|)]$;
15 **return** $\mathcal{P}_{new}$;
16 **end**

As outlined in Algorithm 1, the framework starts with an initial set of $\mathcal{N}$ solutions (line 2 in Algorithm 1). This initial population of solutions could be randomly generated or empty solutions. Then an exploration mechanism is executed based on proper crossover and mutation operators, depending on the problem, to produce new offsprings (line 4). The offspring population $\mathcal{Q}$ is therefore combined with the current population $\mathcal{P}$ forming a new population $\mathcal{P} \cup \mathcal{Q}$ of size $2 \times N$ (line 5). After that, an evaluation process is executed to evaluate for each solution candidate the set of $\mathcal{M}$ objective functions (line 6). Once the evaluation process is terminated, a survival selection process is done to ensure that unfit solutions are eliminated from the population and reduce it back to $\mathcal{N}$ individuals (line 7). The steps 4-7 in Algorithm 1, are repeated until a stop condition (e.g., number of generations) is satisfied.

The selection process, as shown in Algorithm 2, starts by dividing the population into different levels of non-dominated fronts using the fast-non-dominated sorting (NDS) algorithm [14] (line 2 Algorithm 2). Then, the non-dominated fronts are normalized (line 3) using the normalization procedure defined in Section 3.2. Once the solutions are normalized, the first front $\mathbb{F}_1$ is used to estimate the $\mathcal{L}_p$ norm that best fits the geometry of approximated PF (line 4) (More details are described in Section 3.3).

The new population for the next generation is created in lines 5-14 of Algorithm 2. As long as the size of the new population is not reached. Non-dominated Fronts' individuals are assigned a survival score based on the level of the non-dominated front and the estimated p norm. Survival score computation procedure is defined in Algorithm 3.

Solutions from the best non-dominated levels are chosen front-wise (lines 8-11). The goal of this step is to keep good performance solutions (the most advanced solutions). This elitist strategy will allow to converge faster towards the Pareto-optimal Front. However, it is commonly the case that the last front could not be entirely maintained to fit the size ($\mathcal{N}$) of the next generation population. As a consequence, the remaining solutions are chosen according to the descending order of their survival scores (lines 13-14).

We have to mention here that the survival score is computed for the first non-dominated-fronts even though they are chosen front wise. The idea behind is to use survival scores during the reproduction process. In fact, parents are selected for reproduction using the binary tournament selection mechanism: a pair of individuals is randomly selected from the population; the winner of the tournament is the solution with the best non-dominated level or the solution with the largest survival score and the smallest constraints violation at the same level of non-dominated rank.

Details about the principle components of AGE-MOEA++ are described in Sections 3.2, 3.3, and 3.4.

3.2 Normalization

Normalizing the objective space is an important step in multi-objective optimization problems. In fact, objective functions may have different scales, which leads to neglecting one or more objective functions. As a consequence, this impacts how diversity and performance measures of solutions are compared when dominance relations are not sufficient. For that, we suggest normalizing our objective functions by applying the same formula used in [13] and also used in AGE-MOEA:

$$\hat{f}_i(S) = \frac{f_i(S) - Ideal_i}{a_i}, \forall S \in \mathbb{F}, \forall i \in [\![1; M]\!] \tag{5}$$

$$Ideal_i = \min_{\forall S \in \mathbb{F}_1} f_i(S), \forall i \in [\![1; M]\!] \tag{6}$$

$$[z^1, \ldots, z^i, \ldots, z^{\mathcal{M}}]^T \times [1/a_1, \ldots, 1/a_i, \ldots, 1/a_{\mathcal{M}}]^T = 1_{\mathcal{M}} \tag{7}$$

$$z^i = arg\max_{S \in \mathbb{F}_1}(f_i(S) - Ideal_i), \forall i \in [\![1; M]\!] \tag{8}$$

Where $f_i(S)$ denotes the objective f_i for the solution S and $Ideal_i$ is i^{th} component of the ideal point. The ideal point represents the minimum value of the i^{th} objective across all solutions in the first front $\mathbb{F}_1$. The objectives are translated to have the ideal point equals to the origin of the axes. Thereafter, a $\mathcal{M}$-dimensional linear hyperplane Z^{max} is constructed based on the extreme points $(z^i, i \in [\![1; M]\!])$ in each objective axis of the first Front $\mathbb{F}_1$. The denominator a_i is the intercept of the M-dimensional hyperplane with the objective axis f_i, and it is obtained by solving the linear system in equation (7). In case that the system is indefinite or leading to abnormal normalization (e.g., $a_i = 0$), the min-max normalization is used (i.e., a_i is replaced by $z^i_i - Ideal_i$).

3.3 Geometry Fitting

To determine the norm $\mathcal{L}_p$ such that the corresponding unit hypersurface best fits the geometry of normalized objectives, we need to find the value of p that makes all points in $\mathbb{F}_1$ equally distant to the ideal point, which coincides with the origin of the axes $\vec{0}$ after the normalization.

The fitting process consists in solving the following system of non-linear equations:

$$\begin{cases} (f_1(S_1)^p + f_2(S_1)^p + \cdots + f_M(S_1)^p)^{\frac{1}{p}} = 1 \\ (f_1(S_2)^p + f_2(S_2)^p + \cdots + f_M(S_2)^p)^{\frac{1}{p}} = 1 \\ \cdots \\ (f_1(S_q)^p + f_2(S_q)^p + \cdots + f_M(S_q)^p)^{\frac{1}{p}} = 1 \end{cases} \tag{9}$$

where q is the number of points in the front $\mathbb{F}_1$.

Several numerical analysis methods have been proposed to resolve such systems of nonlinear equations (e.g., Newton's iterative method [16], Levenberg- Marquardt algorithm). However, such methods are computationally expensive and not suitable for computing the value of p in each iteration with negligible overhead.

For that reason, we approximate the value of p using the method proposed in [22]. It consists in using the central point of the front $\mathbb{F}_1$ for which the corresponding $\mathcal{L}_p$ exponential equation can be easily computed with an exact method. Moreover, the overall complexity of this method is $O(M \times N)$, which makes it a fast procedure to estimate the geometry of the non-dominated front and can be easily incorporated in the cycle of evolutionary algorithms.

3.4 Survival Score

To select the best individuals that will participate in the next generation, a survival score based on proximity and diversity is calculated as the following:

$$Score(S)_{S \in \mathbb{F}_r} = \begin{cases} Proximity(S) \times Diversity(S), & if\ r = 1 \\ Proximity(S) & \text{Otherwise} \end{cases} \tag{10}$$

$$Proximity(S) = \frac{1}{\|\hat{f}(S)\|_p} \tag{11}$$

$$Diversity(S) = \min_{S,T \in \mathbb{F}_1, S \neq T} \|\hat{f}(S) - \hat{f}(T)\|_p \tag{12}$$

In fact, for each generic solution $S \in \mathbb{F}_r$, the survival score is defined based on the rank of its non-dominated front as illustrated in equation (10). For solutions of the first non-dominated front $\mathbb{F}_1$, the survival score is defined by the combination of the proximity and the diversity scores. While for solutions of the other fronts, the proximity score only is used. The idea behind is to enhance the convergence to the optimal PF by increasing the chance of survival of the most advanced solutions in the rest of the fronts.

For the proximity measure defined in (Eq. (11)), it is evaluated as the inverse of the distance between the solution and the ideal point (see Eq. (6)). The diversity of the solutions $S \in \mathbb{F}_r$ is computed as the p-norm distance to the nearest adjacent solutions in the front $\mathbb{F}_1$. Algorithm 3 details the procedure that assigns survival scores. The survival scores for the first non-dominated front $\mathbb{F}_1$ are computed in lines 2-21, while the score for the other non-dominated fronts $\mathbb{F}_r (r > 1)$ are computed in lines 21-26.

For what concerns $\mathbb{F}_1$, first, all extreme points are assigned the maximum possible survival score ($+\infty$) with the purpose to preserve them in the next generation's population (lines 3-4). This allows to maintain the curvature of the PF during the process of convergence. Then, two sets are initialized: (i) Ω containing all solutions yet to score (line 5), and (ii) $\overline{\Omega}$ keeps track of already scored solutions (line 6). After that, the proximity score for the solutions in Ω is computed according to (Eq.11) (lines 7-9), and the pairwise $\mathcal{L}_p$ distances between all solutions in $\mathbb{F}_1$ are computed in lines 10-14. The survival score is computed within the loop in lines 16-20. In each loop, the method computes the diversity score for a random solution selected based on a stochastic diversity mechanism inspired by K-means++ initialization heuristic (line 17). More details about the method are described in Algorithm 4.

Finally, Survival scores for the solutions in ($\mathbb{F}_r, r > 1$) are calculated as proximity scores (lines 27-30). Hence, solutions closer to the hypersurface generated by $\mathcal{L}_p$ have larger scores.

3.5 Diversity: Problem Formalization and Stochastic Resolution

The problem of diversity promoting in MOP and MaOP, in which the objective is to have a well distributed and a good dispersion of solutions over the obtained PF, can be seen as an instance of the NP-Hard Max-Min dispersion maximization problem [17]. In fact, Max-Min dispersion maximization problem considers a set P of n elements, and a function d that assigns a non-negative real number $d(p, q)$ for each pair of elements $p, q \in P$. The objective is to find a subset $G \subseteq P$ such that the cost(G) = $min\{\ d(p, q)|\ p, q \in G\}$ is maximized [1]. As consequence, our sub-problem of diversity promoting can be formulated as finding the subset $\overline{\Omega} \subseteq \Omega$ such that:

$$\overline{\Omega} = \arg\max_{G \subseteq \Omega}\ Diversity(G) \tag{13}$$

with:

$$Diversity(G) = \ min\ ||S_i, S_j||_p\ \forall\ S_i, S_j \in G \tag{14}$$

To deal with such problems, we suggest to use the fast and simple k-means++ initialization heuristic. The idea consists in building the subset $\overline{\Omega}$ incrementally by choosing random individuals with very specific probabilities. Basically, each solution is selected with probability proportional to its contribution to the overall cost function. Noticeably, we have to mention here that in the specific context of

Algorithm 3: AGE-MOEA++ Survival Score Procedure.

Input: Pool of Non-Dominated-Front $\mathbb{F}_r$, Exponent p of estimated $norm$, r rank of the non-dominated-front;

1 **begin**
2 **if** r == *1* **then**
3 $\Gamma \longleftarrow$ Extreme-Points($\mathbb{F}_r$);
4 Survival-Score[Γ] $\longleftarrow +\infty$;
5 $\Omega \longleftarrow \mathbb{F}_r \backslash \Gamma$;
6 $\overline{\Omega} \longleftarrow \Gamma$;
7 **foreach** $S \in \Omega$ **do**
8 Proximity[S] $\longleftarrow \frac{1}{||f(S)||_p}$;
9 **end**
10 **foreach** $S_1 \in \Omega$ **do**
11 **foreach** $S_2 \in \Omega$ **do**
12 D[S_1, S_2] $\longleftarrow ||f(S_1) - f(S_2)||_p$;
13 **end**
14 **end**
15 W $\longleftarrow \emptyset$;
16 **while** $|\Omega| > 0$ **do**
17 W, S^*, Value, Ω, $\overline{\Omega} \longleftarrow$ Stochastic-Diversity(W, Ω, $\overline{\Omega}$, D);
18 Diversity[S^*] $\longleftarrow$ Value;
19 Survival-Score[S^*] $\longleftarrow$ Diversity[S^*] $\times$ Proximity[S^*];
20 **end**
21 **else**
22 **foreach** $S \in \mathbb{F}_r$ **do**
23 Survival-Score[S] $\longleftarrow \frac{1}{||f(S)||_p}$;
24 **end**
25 **end**
26 **end**

Algorithm 4: AGE-MOEA++ Stochastic Diversity Mechanism.

Input: Points Sampling Probabilities W, Set of initial solutions Ω, Set of selected solutions $\overline{\Omega}$, $\mathcal{L}_p$ Pairwise distance D;
Output: Updated sampling probabilities weights W, Selected solution S^*, Diversity value, Updated set of initial solutions Ω, Updated set of selected solutions $\overline{\Omega}$;

1 **begin**
2 **if** $W = \emptyset$ **then** // Case Weights Not Initialized
3 **foreach** $S \in \Omega$ **do** // Uniform Probability Weights
4 W[S] $\longleftarrow \frac{1}{|\Omega|}$;
5 **end**
6 **end**
7 $S^* \longleftarrow$ Random-Solution(Ω, W[Ω]);
8 $\Omega \longleftarrow \Omega \backslash S^*$;
9 Diversity[S^*] $\longleftarrow$ min D[S^*, $\overline{\Omega}$];
10 $\overline{\Omega} \longleftarrow \overline{\Omega} \cup \{S^*\}$;
11 **foreach** $S \in \Omega$ **do**
12 W[S] $\longleftarrow \frac{\min_{V \in \overline{\Omega}} D[V,S]}{\sum_{T \in \Omega} \min_{V \in \overline{\Omega}} D[V,T]}$;
13 **end**
14 **return** W, S^*, *Diversity*, Ω, $\overline{\Omega}$;
15 **end**

MOEA, we need to score all the individuals of the population. As a consequence, the problem could be reformulated as finding the best selection order that maximizes the overall cost function.

For what concerns diversity score evaluation, Algorithm 4 details the mechanism used to assign diversity score for each individual (solution) of the initial population Ω. In fact, for the first individual, a uniform probability (lines 2-6 of Algorithm 4) is used to choose a point at random from Ω (line 7). After that, the procedure defines its diversity score as the minimum $\mathcal{L}_p$ distance with regards to the solutions in $\overline{\Omega}$ (line 9). $\overline{\Omega}$ is the set of scored solutions seted initially with extreme points. For the next individuals' selection, we choose a solution based on a probability proportional to its contribution to the overall diversity by calculating the ratio between the diversity gain for each solution and the sum of the gains (lines 11-13). Each time, the diversity of a solution S is computed with regards to solutions that have already been selected ($\overline{\Omega}$) in the previous iterations rather than considering all solutions in $\mathcal{F}_1$. As consequence, this allows to well characterize the diversity of solutions.

4 EMPIRICAL STUDY

This section reports the conducted experiments to show the effectiveness and validity of AGE-MOEA++ to solve multi and many objective optimization problems based on different literature test problems.

In order to evaluate our approach, we answer the following questions:

(1) **How does AGE-MOEA++ balance between convergence and diversity in multiobjective optimization context?** To investigate this, we propose experiments where state-of-the-art methods NSGA-II [14], NSGA-III [13], MOEA/D [28], AGE-MOEA [22] are compared to AGE-MOEA++ using literature based multiobjective test problems.
(2) **How does our approach scale in terms of the number of objectives?** To find out, we conduct experiments on high dimensional objective spaces to assess the scalability of our approach.

4.1 Experimentation Environment Settings

Table 1: Experiments shared parameters. n denotes number of decision variables.

Parameters	$\mathcal{M} = 3$	$\mathcal{M} = 5$	$\mathcal{M} = 10$
Population Size [13]	91	210	275
Number of fitness evaluation [22]	27300	63000	82500
PF Evaluation (IGD, GD) Size	2016	4845	5005
Number of Generations		300	
Polynomial Mutation Probability [22]		$p_m = 1/n$	
Mutation Distributed Index [22]		$\eta_m = 20$	
SBX Probability [11]		$p_c = 1$	
SBX Distributed Index [11]		$\eta_c = 30$	

We implemented AGE-MOEA++[1] in Python using Pymoo [6]. Pymoo provides the source code for all benchmark problems as well as the algorithms we use as baselines in our study. For all MOEAs, we used the same parameter setting reported in the related literature [13, 14, 22, 28]. Table 1 shows all parameters' values for the evaluated MOEAs. For all the other parameters we use the values suggested by their developers. To ensure a fair comparison, we use the same population size, the same number of fitness evaluations, and the same size of optimal PF for all algorithms in our study. In particular, we set the population size N=91, 210, and 275 for the number of objectives M=3, 5, and 10 respectively. The number of fitness evaluations is set to $N \times 300$ iterations.

To assess the effectiveness of AGE-MOEA++ in Multi and Many-Objective Problems (MaOPs) context, we considered DTLZ [15] test benchmarks, with the number of objectives $\mathcal{M}$= 3, 5, and 10. It contains several test problems with different properties, such as concave (e.g., DTLZ2), convex (e.g., Convex DTLZ2), disconnected (e.g., DTLZ7), and degenerate [19] (e.g., DTLZ5) PFs. Therefore, such a suite is a good representation of various real-world scenarios. Fig. 1 illustrates the curvature of some DTLZ test problems used in our experiments.

For each test problem, we run each algorithm 30 times to account for their non deterministic nature. In each independent run, we collected the solutions produced by a given algorithm at the end of the search and computed:

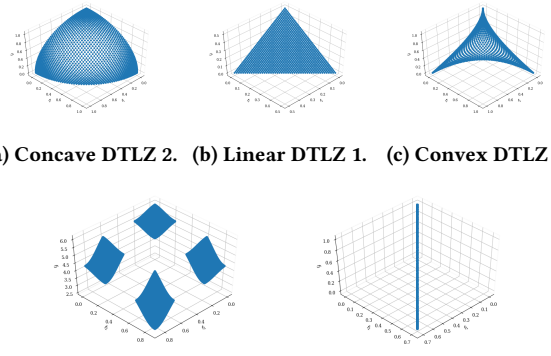
- The inverted generational distance (IGD) [7] to measure its overall quality. In fact, the IGD measures the distance from any point in optimal PF to the closest point in obtained PF. As consequence, it provides a single scalar value measuring both proximity and diversity of the obtained solutions.
- The GD performance indicator [26] to evaluate the convergence of the algorithms because it measures the distance from obtained solutions to the optimal PF. The smaller the IGD and GD values, the better the performance of the algorithm.

To evaluate the significance of the differences among the different MOEAs, we use the Wilcoxon rank-sum test [8] with significance level $\alpha = 0.05$. A significant p-value (p-value $< \alpha$) indicates that an algorithm $\mathcal{A}$ achieves significant performance in terms of indicator measures than another algorithm $\mathcal{B}$ across 30 runs for a given test problem.

4.2 Empirical Results

In this subsection, we present and discuss the results of the comparison between AGE-MOEA++ and the four MOEAs baselines. Table 2 and 3 summarize the number of benchmark problems for which AGE-MOEA++ significantly outperforms a baseline MOEA and vice versa based on the IGD and GD measures respectively. Table 4 provides the average (mean) and the standard deviation (shown between parentheses) of the IGD achieved by each MOEA across 30 independent runs. The table also reports whether a baseline (e.g., MOEA/D) statistically outperforms (denoted with ↑), or is statistically equivalent (≈) to AGE-MOEA++ according to the Wilcoxon rank-sum test. Each time, the best performance is highlighted in blue color.

[1]The source codes are available at https://github.com/AdwLab/AGE-MOEA-plus-plus



(a) Concave DTLZ 2. **(b) Linear DTLZ 1.** **(c) Convex DTLZ 2.**

(d) Disconnected DTLZ 7. **(e) Degenerate DTLZ 5.**

Figure 1: Curvature of Used DTLZ test problems.

From Tables 2-3, AGE-MOEA++ performs significantly better than AGE-MOEA in 14 out of 18 DTLZ test problems based on the IGD and the GD measures. Table 4 illustrates that the IGD values achieved by AGE-MOEA ++ are sometimes one order of magnitude smaller than the IGD values achieved by AGE-MOEA. For example, for Convex DTLZ 2 and $\mathcal{M} = 5$, AGE-MOEA++ obtains an average IGD value equal to 7.354e(-2) compared to an IGD value of 7.703e(-1) achieved by AGE-MOEA. On the other hand, AGE-MOEA is statistically equivalent to AGE-MOEA++ in two test problems (DTLZ3 and DTLZ7 with $\mathcal{M} = 3$). For the 2 remaining tests, no significant difference is observed. This is due to the fact that AGE-MOEA++ stochastic method has the capacity to correct bad choices, especially in the first EA generations, which gives it an advantage over AGE-MOEA deterministic approach.

For what regards NSGA-II, we can notice that AGE-MOEA ++ outperforms NSGA-II in 13 out of 18 problems based on the IGD and 14 out of 18 problems based on the GD measure. For some problems, the differences between the two MOEAs are above two orders of magnitude. For example, for DTLZ2 test problem and $\mathcal{M} = 10$ (Many Objective Problems), AGE-MOEA ++ obtains an average IGD value equals to 6.156e(-1) compared to an IGD value of 1.25e(+1) for NSGA-II. On the other side, NSGA-II performs significantly better than AGE-MOEA ++ in one test problem: DTLZ 4 for $\mathcal{M} = 3$. For the remaining 3 test problems, there is no significant difference. This confirms the result which shows that NSGA-II is not suitable for many objective optimization problems.

From Tables 2-3, AGE-MOEA++ achieves significantly lower (better) values than NSGA-III in 14 out of 18 problems based on IGD indicator and 12 out of 18 based on the GD measure. In particular, NSGA-III produces globally significantly higher (worst) IGD values for the majority of test problems. The most significant difference is observed for DTLZ 4 with $\mathcal{M} = 10$ where AGE-MOEA++ obtains an average IGD value equals to 4.913e(-1) while NSGA-III achieves an IGD value of 1.71e(+0). This is owing to the fact that the method used to generate reference lines is NSGA-III is not adaptable to the shape of the PF.

Finally, from the comparison between AGE-MOEA++ and MOEA/D, we can observe from Tables 2-3, the former significantly outperforms the latter in the majority of test problems (13/18 based

on IGD and 11/18 based on the GD). The largest difference between the two MOEAs is observed for the concave DTLZ 2 test problem with $\mathcal{M} = 3$. In this case, AGE-MOEA++ has an average IGD value equals to 8.472e(-2) while MOEA/D has an IGD value of 7.397e(-1). In particular, for $\mathcal{M} = 10$, MOEA/D outperforms AGE-MOEA++ based on DTLZ1, DTLZ2, and DTLZ3 test problems. This is due to the shape of these Pareto fronts (Continuous linear and concave PF) which makes it advantageous through the use of reference lines.

Table 2: Number of DTLZ benchmarks [15] in which an algorithm $\mathcal{A}$ (e.g., AGE-MOEA++) statistically outperforms (<) another algorithm $\mathcal{B}$ (e.g., MOEA/D) according to the Wilcoxon test (p-value ≤ 0.05) Based On the IGD measure.

Comparaison	$\mathcal{M} = 3$	$\mathcal{M} = 5$	$\mathcal{M} = 10$
AGE-MOEA++ < AGE-MOEA	5	5	4
AGE-MOEA < AGE-MOEA++	0	0	0
AGE-MOEA++ < NSGA-III	5	5	4
NSGA-III < AGE-MOEA++	0	0	0
AGE-MOEA++ < MOEA/D	7	4	2
MOEA/D < AGE-MOEA++	0	0	3
AGE-MOEA++ < NSGA-II	4	5	4
NSGA-II < AGE-MOEA++	1	0	0

5 CONCLUSION

In this paper, we propose a new MOEA, called AGE-MOEA++, that inherits from the computational fast and elitist framework NSGA-II, incorporates K-means++ initialization heuristic to obtain distributed and diverse set of solutions while reducing the diversity process complexity and uses the AGE-MOEA Pareto Front geometry estimation technique to be adapted to different shapes of the PF. Our solution shows better results than the state-of-the-art approaches in terms of IGD and is competitive in terms of GD. As future works, we aim to work on the integration of DM's preferences in terms of different subsets of the PF in order to build an efficient optimization/recommendation system for different combinatorial problems such as Advertising Campaigns Allocation Problem.

Table 3: Number of DTLZ benchmarks [15] in which an algorithm $\mathcal{A}$ (e.g., AGE-MOEA++) statistically outperforms (<) another algorithm $\mathcal{B}$ (e.g., MOEA/D) according to the Wilcoxon test (p-value ≤ 0.05) Based On the GD measure.

Comparaison	$\mathcal{M} = 3$	$\mathcal{M} = 5$	$\mathcal{M} = 10$
AGE-MOEA++ < AGE-MOEA	5	5	4
AGE-MOEA < AGE-MOEA++	0	0	0
AGE-MOEA++ < NSGA-III	4	4	4
NSGA-III < AGE-MOEA++	0	1	0
AGE-MOEA++ < MOEA/D	8	2	1
MOEA/D < AGE-MOEA++	0	3	4
AGE-MOEA++ < NSGA-II	5	5	4
NSGA-II < AGE-MOEA++	1	0	0

ACKNOWLEDGMENTS

This work is funded by the ANRT CIFRE Program (2019/0877).

REFERENCES

[1] Toshihiro Akagi, Tetsuya Araki, Takashi Horiyama, Shin-ichi Nakano, Yoshio Okamoto, Yota Otachi, Toshiki Saitoh, Ryuhei Uehara, Takeaki Uno, and Kunihiro Wasa. 2018. Exact algorithms for the max-min dispersion problem. In *International Workshop on Frontiers in Algorithmics*. Springer, 263–272.

[2] David Arthur and Sergei Vassilvitskii. 2006. *k-means++: The advantages of careful seeding*. Technical Report. Stanford.

[3] Olivier Bachem, Mario Lucic, S Hamed Hassani, and Andreas Krause. 2016. Approximate k-means++ in sublinear time. In *Thirtieth AAAI conference on artificial intelligence*.

[4] Johannes Bader and Eckart Zitzler. 2011. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation* 19, 1 (2011), 45–76.

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

[6] Julian Blank and Kalyanmoy Deb. 2020. pymoo: Multi-objective optimization in python. *IEEE Access* 8 (2020), 89497–89509.

[7] Carlos A Coello Coello and Margarita Reyes Sierra. 2004. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican international conference on artificial intelligence*. Springer, 688–697.

[8] William Jay Conover. 1999. *Practical nonparametric statistics*. Vol. 350. john wiley & sons.

[9] Indraneel Das and John E Dennis. 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization* 8, 3 (1998), 631–657.

[10] Kalyanmoy Deb. 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 3–34.

[11] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. 1995. Simulated binary crossover for continuous search space. *Complex systems* 9, 2 (1995), 115–148.

[12] Kalyanmoy Deb, Shamik Chaudhuri, and Kaisa Miettinen. 2006. Towards estimating nadir objective vector using evolutionary approaches. In *GECCO'06*. 643–650.

[13] Kalyanmoy Deb and Himanshu Jain. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* 18, 4 (2014), 577–601.

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

[15] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2002. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, Vol. 1. IEEE, 825–830.

[16] John E Dennis Jr and Robert B Schnabel. 1996. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM.

[17] Erhan Erkut. 1990. The discrete p-dispersion problem. *European Journal of Operational Research* 46, 1 (1990), 48–60.

[18] Xiaoyu He, Yuren Zhou, Zefeng Chen, and Qingfu Zhang. 2018. Evolutionary many-objective optimization based on dynamical decomposition. *IEEE Transactions on Evolutionary Computation* 23, 3 (2018), 361–375.

[19] Hisao Ishibuchi, Hiroyuki Masuda, and Yusuke Nojima. 2016. Pareto Fronts of Many-Objective Degenerate Test Problems. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 807–813. https://doi.org/10.1109/TEVC.2015.2505784

[20] Hisao Ishibuchi, Takashi Matsumoto, Naoki Masuyama, and Yusuke Nojima. 2020. Many-objective problems are not always difficult for Pareto dominance-based evolutionary algorithms. In *ECAI 2020*. IOS Press, 291–298.

[21] Hisao Ishibuchi, Noritaka Tsukamoto, Yasuhiro Hitotsuyanagi, and Yusuke Nojima. 2008. Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In *GECCO'08*. 649–656.

[22] Annibale Panichella. 2019. An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, Anne Auger and Thomas Stützle (Eds.). ACM, 595–603.

[23] Hiroyuki Sato, Hernán E Aguirre, and Kiyoshi Tanaka. 2007. Controlling dominance area of solutions and its impact on the performance of MOEAs. In *International conference on evolutionary multi-criterion optimization*. Springer, 5–20.

[24] Yanan Sun, Gary G. Yen, and Zhang Yi. 2019. IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Trans. Evol. Comput.* 23, 2 (2019), 173–187.

[25] Anthony C Thompson and Anthony C Thompson. 1996. *Minkowski geometry*. Cambridge University Press.

Table 4: IGD values achieved by our algorithms and baselines on DTLZ benchmark with $\mathcal{M}$ =3, 5, and 10 objectives.

Problem	$\mathcal{M}$	MOEA/D	NSGA-III	AGE-MOEA	NSGA-II	AGE-MOEA++
DTLZ1	3	7.3784e+2 (5.05e+1)	2.484e+2(2.57e+1)	2.634e+2(3.47e+1)	3.05e+2(3.35e+1)	2.15e+2(2.53e+1)
DTLZ2	3	7.397e-1(7.15e-2)	1.148e-2(1.23e-1)	9.477e-2(5.5e-3)	1.036e-1(5.3e-3)	8.472e-2(4.41e-3)
C-DTLZ2	3	1.755e-1(3.14e-2)	1.109e-1(1.96e-2)	7.26e-2(7.64e-3)	1.106e-1(1.04e-2)	6.26e-2(4.44e-3)
DTLZ3	3	2.26e+3(8.04e+1)	7.027e+2(5.30e+	≈6.51e+2(8.11e+1)	7.80e+2(9.11e+1)	6.87e+2(5.36e+1)
DTLZ4	3	6.046e-1(4.67e-2)	3.28e-1(2.81e-1)	5.37e-1(3.21e-1)	↑1.08e-1(7.08e-3)	4.88e-1(3.03e-1)
DTLZ5	3	5.985e-1(7.23e-2)	3.668e-2(4.39e-3)	4.202e-2(3.24e-3)	≈2.999e-2(3.884e-3)	3.439e-2(4.70e-3)
DTLZ6	3	8.028e+1(5.93e-1)	5.931e+1(8.45e-1)	6.195e+1(7.15e-1)	5.762e+1(1.31e+0)	5.742e+1(1.19e+0)
DTLZ7	3	7.117e-1(8.60e-2)	4.126e-1(4.30e-2)	≈ 2.652e-1(5.53e-2)	3.284e-1(6.55e-2)	3.22e-1(6.67e-2)
DTLZ1	5	≈2.963e+2(3.10e+1)	9.746e+2(5.41e+1)	9.066e+2(8.90e+1)	2.063e+3(1.07e+2)	3.162e+2(2.80e+1)
DTLZ2	5	4.084e-1(3.25e-2)	2.647e-1(1.36e-2)	3.215e-1(1.02e-2)	1.991e+0(1.59e-1)	2.183e-1(7.17e-3)
C-DTLZ2	5	2.341e-1(1.08-2)	1.321e-1(2.19e-2)	7.703e-1(1.72e-1)	7.186e+1(7.35e-2)	7.354e-2(3.70e-1)
DTLZ3	5	1.164e+3(9.66e+1)	1.63e+3(9.38e+1)	1.39e+3(1.58e+2)	4.85e+3(3.92e+2)	9.31e+2(8.08e+1)
DTLZ4	5	5.453e-1(1.45e-1)	2.861e-1(1.85e-2)	4.39e-1(2.55e-2)	5.455e+00(5.55e-1)	2.66e-1(1.62e-2)
DTLZ1	10	↑7.579e+1(9.87e+0)	1.191e+3(8.44e+1)	2.497e+3(9.62e+1)	2.63e+3(1.30e+2)	6.95e+2(7.36e+1)
DTLZ2	10	↑4.22e-1(8.61e-2)	9.72e-1(9.32e-2)	9.28e+0(1.91e+0)	1.25e+1(2.20e+0)	6.156e-1(2.23e-2)
C-DTLZ2	10	3.603e-1(2.11e-3)	3.679e-1(1.42e-1)	4.209e+1(6.52e+0)	2.19e+2(5.04e+1)	1.109e-1(3.53e-3)
DTLZ3	10	↑ 5.171e+2(9.5e+1)	5.353e+3(2.35e+2)	1.101e+4(7.91e+2)	1.008e+4(7.14e+2)	2.44e+3(2.43e+2)
DTLZ4	10	5.127e-1(8.07e-2)	1.71e+0(1.12e-1)	4.761e+0(8.32e-1)	7.421e+0(3.68e-1)	4.913e-1(3.53e-2)

[26] David Allen Van Veldhuizen. 1999. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations.* Air Force Institute of Technology.

[27] Yi Xiang, Yuren Zhou, Xiaowei Yang, and Han Huang. 2019. A many-objective evolutionary algorithm with Pareto-adaptive reference points. *IEEE Transactions on Evolutionary Computation* 24, 1 (2019), 99–113.

[28] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[29] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation* 3, 4 (1999), 257–271.