An Adaptive Evolutionary Algorithm based on Non-Euclidean Geometry for Many-objective Optimization

Annibale Panichella
Delft University of Technology, The Netherlands
a.panichella@tudelft.nl

ABSTRACT

In the last decade, several evolutionary algorithms have been proposed in the literature for solving multi- and many-objective optimization problems. The performance of such algorithms depends on their capability to produce a well-diversified front (diversity) that is as closer to the Pareto optimal front as possible (proximity). Diversity and proximity strongly depend on the geometry of the Pareto front, i.e., whether it forms a Euclidean, spherical or hyperbolic hypersurface. However, existing multi- and many-objective evolutionary algorithms show poor versatility on different geometries. To address this issue, we propose a novel evolutionary algorithm that: (1) estimates the geometry of the generated front using a fast procedure with $O(M \times N)$ computational complexity (M is the number of objectives and N is the population size); (2) adapts the diversity and proximity metrics accordingly. Therefore, to form the population for the next generation, solutions are selected based on their contribution to the diversity and proximity of the non-dominated front with regards to the estimated geometry. Computational experiments show that the proposed algorithm outperforms state-of-the-art multi and many-objective evolutionary algorithms on benchmark test problems with different geometries and number of objectives (M=3,5, and 10).

CCS CONCEPTS

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

KEYWORDS

Many-objective optimization, Genetic Algorithms, norms, Non-Euclidean geometry

ACM Reference Format:
Annibale Panichella. 2019. An Adaptive Evolutionary Algorithm based on Non-Euclidean Geometry for Many-objective Optimization. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3321707.3321839

1 INTRODUCTION

Evolutionary algorithms have been widely used in literature to solve multi and many-objective optimization problems (MOPs), where the objectives are conflicting with each other. The goal of multi and many-objective evolutionary algorithms (MOEAs) is to approximate the optimal Pareto front (PF) with a set of non-dominated solutions. To reach this goal, multi and many-objective evolutionary algorithms (MOEAs) try to generate non-dominated fronts that are as close to PF as possible (*proximity* or *convergence*), and that is well-distributed over the optimal PF (*diversity*).

In NSGA-II, the proximity is achieved using the Pareto-dominance and diversity is preserved using the crowding distance [7]. `MOEA/D` maintains the diversity by using well-diverse weight vectors, and it tries to reach the PF using the aggregation function-based selection [12]. While multi-objective EAs perform well for MOPs with $M \leq 3$, their performance degrades when handling many-objective problems [13, 14], i.e., MOPs with more than three objectives. This degradation happens because the number of solutions that are non-dominated in each population increases exponentially with the number of objectives M. This leads to a dramatic loss of the selection pressure toward the PF [13].

To overcome such limitations, researchers have proposed different strategies to increase either the proximity or the diversity of the population. These strategies can be classified into three categories: (1) reference-point based, (2) dominance-relation based, and (3) indicator based strategies. For example, `NSGA-III` replaces the crowding distance with a reference-point based operator [6, 10], which promotes both proximity and population member diversity. θ-`DEA` [22] uses a new dominance relation, called θ-dominance, which has a stronger selection pressure than the Pareto-dominance. `GrEA` [20] uses the *grid dominance* and *grid difference* to strengthen the selection pressure and maintaining population members diversity using a grid-based strategy. In `AR-MOEA` [16], the selection criterion is based on the *enhanced inverted GD* (IGD-NS) indicator [16]. To calculate this indicator, `AR-MOEA` maintains a set of reference points that are adaptively maintained and updated.

The MOEAs mentioned above use strategies and heuristics that are built upon the implicit assumption that the PF (or its non-dominated front approximation) has a Euclidean geometry. For example, in `NSGA-III` and `AR-MOEA`, the reference points are generated using Das and Dennis's systematic approach [4], which places points on a normalized hyper-plane (a flat hypersurface). `GrEA` divides the objective space in grids of equal size. However, in many MOPs the PF is convex (i.e., hyperbolic geometry) or concave (i.e., spherical geometry) [11]. For these reasons, recent studies [11, 19] applied non-linear fitting methods to model the shape of the front generated by MEOAs. However, front modelling techniques incur in a substantial overhead to due the computation complexity of the applied fitting methods.

In this paper, we introduce a novel MOEA, called `AGE-MOEA` (Adaptive Geometry Estimation based MOEA), for evolutionary

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321839

multi and many-objective optimization. The proposed framework does not make any assumption about the geometry of the PF and its approximation (non-dominated front) generated in each generation. AGE-MOEA inherits the overall framework of NSGA-II, but it replaces the crowding distance with a *survival score* that combines both diversity and proximity of the non-dominated fronts. First, AGE-MOEA uses a fast heuristic with $O(M \times N)$ computational complexity[1] to estimates the geometry of the front in each generation. Then, proximity is computed as the distance between each population member and the ideal point while the diversity is measured using the distance among population members. The distance used to compute proximity and diversity corresponds to the L_p norm that is associated with the estimated geometry.

To assess the performance of the proposed AGE-MOEA, we used a benchmark of 15 test problems that have PFs with different shapes and geometries and varying the number of objectives $M \in \{3, 5, 10\}$. In particular, we use the Maf test suite [2] used in previous empirical studies to assess the performances on MOEAs (e.g., [9, 18]). Experimental results show that AGE-MOEA is a versatile MOEA that achieves promising results on MOPs with concave, convex, disconnected, inverted and degenerate PFs. Besides, AGE-MOEA outperforms five state-of-the-art MOEAs, namely NSGA-III, θ-DEA, GrEA, MOEA/D, and AR-MOEA.

The remainder of the paper is organized as follows. Section 2 introduces basic concepts in multi and many-objective optimization. It also describes the L_p norms and their connection with hypersurfaces with difference geometries (curvatures). Section 3 details the proposed framework AGE-MOEA. Section 4 describes the empirical studies and results. Finally, Section 5 concludes the paper.

2 BACKGROUND

A multi-objective optimization problem can be formulated as:

$$min f(x) = (f_1(x), f_2(x), \ldots, f_M(x)]^T \quad (1)$$
$$\text{subject to } x \in U \subseteq \mathbb{R}^n$$

where $x = (x_1, \ldots, x_n)^T$ is a n-dimensional solution (or decision) vector from the space $U \subset \mathbb{R}^n$; $f : U \rightarrow \mathbb{R}^M$ is an objective function vector that consists of M contrasting objective functions (or simply objectives). A multi-objective problem with $M > 3$ is often referred to as *many-objective* problem.

A solution x *dominates* another solution y (denoted by $x \prec y$), if $f_i(z) \leq f_i(y)\ \forall i \in 1, \ldots, M$ and there exists at least one index $j \in 1, \ldots, M$ such that $f(x) < f(y)$. Moreover, a solution x^* is *Pareto optimal* if there exists no other solution $y \in U$ such that $y \prec x^*$. The objectives of a Pareto optimal solution x^* form the so called *Pareto optimal objective vector* and it is denoted by $f(x^*)$. The set of all Pareto optimal solutions in U is called the *Pareto optimal solution set* (PS) and the set of the corresponding objective vectors is called the *Pareto front* (PF) or Pareto frontier.

2.1 L_p Norms and Non-Euclidean Geometry

In the M-dimensional space $\mathbb{R}^M$, the length (or *norm*) of a vector $v = (v_1, \ldots, v_M)$ is computed using the Euclidean formula:

$$\|v\|_2 = \left(v_1^2 + \cdots + v_M^2\right)^{1/2} \quad (2)$$

[1] M is the number of objectives and N is the population size.

In the Euclidean space, the distance between two points A and B is the norm of the straight line connecting the two points, i.e., $d(A, B) = \|A-B\|_2$. However, the Euclidean norm does not necessary provide the most accurate measure of the distance between two points in a generic M-dimensional space [15].

An L_p norm (or p-norm) is a generalization of the Euclidean norm and it is defined by the formula [15]:

$$\|v\|_p = \left(v_1^p + \cdots + v_M^p\right)^{1/p} \quad (3)$$

The Euclidean norm (L_2 norm or 2-*norm*) represents a special case of the L_p norm where $p = 2$.

Different values of p correspond to different measures of the distance between two points A and B in $\mathbb{R}^M$. Indeed, the set of points $v \in \mathbb{R}^M$ that are equidistant to a common point (e.g., the origin of the axes) varies depending on the used L_p norm [15]. The set of all points with a distance $\|\cdot\|_p = 1$ to the origin of the axes forms a unit hypersurface[2]. The geometry (curvature) of the hypersurface strictly depends on the value of the exponent p [15].

To better explain this aspect, let us consider the unit hypersurfaces for a bi-dimensional space (in this case we have unit curves). 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 (with distance one) 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 radius equal to one. Instead, for $p < 1$, the equidistant points lie on a hyperbolic curve. Therefore, the value of p determines the geometry (curvature) of the unit hypersurface associated with the norm L_p.

Previous studies suggested to leverage the relationship between L_p and the curvature of the corresponding unit hypersphere to estimate the shape of the non-dominated front produced by a MOEA. Aguirre et al. [11] proposed an indicator-based MOEA that uses a family of L_p curves to construct an appropriate reference set, which is used to compute the Δ_p indicator. Recently, Tian et al. [19] used a generalized variant of the Lp norm and used the Levenberg-Marquardt (LM) algorithm to derive the parameters of the generalized simplex model that minimize the fitting error. However, methods used in prior studies to model the generated fronts are particularly expensive. Front modelling requires to solve a non-linear fitting problem with iterative numerical methods, leading to a large computation overhead. For example, the fitting process in [19] has an overall complexity of $O(G'M2(M + N))$, where G' is the number of iterations of the LM algorithm.

This paper introduces a fast procedure to estimate the geometry of the non-dominated front, whose overall complexity is $O(M \times N)$, where M is the number of objectives and N is the population size. Besides, we incorporate this fast procedure into NSGA-II and replace the *crowding distance* with *proximity* and *diversity* heuristics.

3 THE PROPOSED ALGORITHM

The framework of the proposed MOEA, namely AGE-MOEA, modifies NSGA-II by replacing the fitness (scores) assigned to the solutions in each non-dominated front. In AGE-MOEA, the crowding distance of NSGA-II is replaced by a *survival score* that combines both *diversity* and *proximity* of the solutions within the same non-dominated

[2] Unit circle in the space $\mathbb{R}^2$ and unit sphere in $\mathbb{R}^3$ for L_2.

Algorithm 1: AGE-MOEA

```
Input: M: Number of objectives
       N: Population size
Result: Final population P
1  begin
2      P ⟵ RANDOM-POPULATION(N)
3      while not (stop_condition) do
4          Q ⟵ GENERATE-OFFSPRING(P)
5          𝔽 ⟵ FAST-NONDOMINATED-SORT(P ⋃ Q)
6          𝔽 ⟵ NORMALIZE(𝔽)
7          p ⟵ GET-GEOMETRY(𝔽_1, M)        /* Equation 8 */
8          d ⟵ 1                /* First non-dominated rank */
9          while | P | + | 𝔽_d | ⩽ N do
10             SURVIVAL-SCORE(𝔽_d, d, p)
11             P ⟵ P ⋃ 𝔽_d
12             d ⟵ d + 1
13         SORT(𝔽_d)                    /* by survival scores */
14         P ⟵ P ⋃ 𝔽_d[1 : (N − |P|)]
15     return P
```

front. First, we present an overview of the proposed framework in Section 3.1. Then, we detail the key ingredients of AGE-MOEA in Sections 3.2, 3.3, and 3.4.

3.1 Overview

Algorithm 1 outlines the proposed framework. The search starts with an initial set of N randomly generated solutions (line 2 in Algorithm 1). Steps 3-14 are repeated until the stop condition (e.g., the number of iterations) is satisfied. In line 4, new solutions (*offspring*) are generated using *crossover* and *mutation*. In this paper, we use the same genetic operators applied in [6, 10], namely the *simulated binary crossover* (SBX) and the *polynomial mutation*. However, other operators could be used as well. The offspring population Q is therefore combined with the current population P forming a new population $Q \bigcup P$ of size $2 \times N$. After that, the obtained population is divided into non-dominated levels (or fronts) using the *non-dominated sorting* algorithm [7]. Then, the non-dominated fronts are normalized using a variant of the normalization procedure defined in NSGA-III [10] (see Section 3.2).

After normalization, the new population for the next generation is created in lines 9-14 of Algorithm 1. Solutions in the first non-dominated front ($\mathbb{F}_1$) are assigned a survival score that combines both *diversity* and *proximity* using the procedure SURVIVAL-SCORE (line 10 of Algorithm 1), which is detailed in Algorithm 2 and described in Section 3.4. In AGE-MOEA, diversity and proximity are computed using the L_p norm, whose exponent p depends on geometry of the optimal Pareto front. Since the optimal front is not known a priori, p is computed based on the geometry of the first non-dominated front in each generation (line 7). In particular, p is approximated using the heuristic described in Section 3.3. Therefore, the L_p norm (as well as the diversity and proximity metrics) is adapted iteratively. The closer the first non-dominated front to the optimal front, the more accurate the estimation of the geometry (i.e., the calculate value of the exponent p) of the PF.

Finally, the new population of M solutions is formed by selecting the solutions from the non-dominated fronts, one front (or level) at a time. Therefore, the solutions from the first front $\mathbb{F}_1$ are selected first, followed by $\mathbb{F}_2$, and so on. The procedure terminates when adding the solutions of the current non-dominated front $\mathbb{F}_d$ would exceed M. In this case, AGE-MOEA selects the remaining solutions from $\mathbb{F}_d$ according to the descending order of their survival scores (lines 13-14 of Algorithm 1).

The survival scores play an important role during reproduction as well. Indeed, parents are selected from the current population P using the *binary tournament selection*: a pair of solutions is randomly selected from P; the winner of the tournament is the solution with the best non-dominated rank (or level) or the solution with the largest *survival score* at the same level of non-dominated rank.

3.2 Normalization

The first non-dominated front $\mathbb{F}_1$ is rescaled and normalized by applying the same formula used in NSGA-III [10]:

$$f_i^n(S) = [(f_i(S) - z_i^{\min}]/a_i \quad \forall S \in \mathbb{F}_1 \tag{4}$$

where $f_i(S)$ denotes the objective f_i for the solution S and $z_i^{\min}$ is the minimum value of the i-th objective across all solutions in the front $\mathbb{F}_1$. With the numerator, the objectives are translated to have the ideal point equal to the origin of the axes. The denominator a_i is the intercept of the M-dimensional hyperplane with the objective axis f_i. The M-dimensional hyper-plane is composed by the extreme vectors $z_i^{\max} = \max_{S \in \mathbb{F}_1}(f_i(S) - z_i^{\min})$, i.e., the largest objectives values in $\mathbb{F}_1$ after the translation toward the origin of the axes. The intercepts are obtained by solving the system of linear equation $Z^{\max}\ a = 1$, where $Z^{\max}$ is the M-dimensional hyper-plane formed by the extreme points of $\mathbb{F}_1$.

As notices by previous studies [21], the system $Z^{\max}\ a = 1$ may be indefinite, impossible or leading to *abnormal normalization results*. In these cases, AGE-MOEA uses the *min-max normalization* where the denominator in Equation 4 is computed as $a_i = (z_i^{\max} - z_i^{\min})$, i.e., the difference between the maximum and the minimum values for the objective f_i across all solutions in the front $\mathbb{F}_1$.

The solutions in all other non-dominated fronts $\mathbb{F}_d$ (with $d > 1$) are scaled and normalized based on the values $z_i^{\min}$ and a_i computed for the first non-dominated front. Therefore, while the objectives for $\mathbb{F}_1$ take values in $[0, 1]$, the objectives for the other non-dominated front $\mathbb{F}_{d>1}$ can have values higher than one.

3.3 The Geometry of the First Front

Determining the geometry for the first non-dominated front $\mathbb{F}_1$ is a *fitting* problem: we need to find the unit hypersurface that best fits the normalized objectives of $\mathbb{F}_1$. The geometry (curvature) of the unit hypersurface is determined by its associated L_p norm. Therefore, we have to find the value p for the norm L_p such that the corresponding unit hypersurface best fits the normalized objectives of the front. An optimal fitting hypersurface has an L_p norm such that all points in $\mathbb{F}_1$ are equally distant to the ideal point $Z^{\min} = (z_1^{\min}, \ldots, z_M^{\min})$, which coincides with the origin of the axes $\vec{0}$ after the normalization (i.e., $z_i^{\min} = 0,\ \forall f_i^n$).

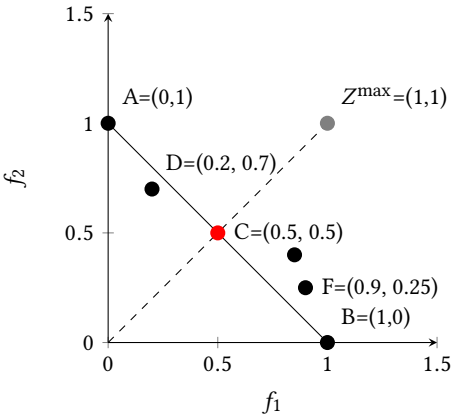


Figure 1: Example of central point C of a normalized non-dominated front.

Furthermore, the extreme points in the front correspond to the intersections of the front with the objective axes. After normalization, the intersection point of $\mathbb{F}_1$ with the axis f_i^n has the objective $f_i^n = 1$ and all other objectives $f_j^n = 0$ for $j \neq i$. Furthermore, its distance to the origin of the axes (ideal point) is always equal to one whatever p exponent is chosen. This is because an extreme point E with coordinate $(f_1^n(E) = 0, \ldots, f_i^n(E) = 1, \ldots, f_M^n(E) = 0)$ has an L_p norm equal to:

$$\begin{aligned} \|E - \vec{0}\|_p &= \left(f_1^n(E)^p + \cdots + f_i^n(E)^p + \cdots + f_1^M(E)^p\right)^{1/p} \\ &= \left(0^p + \cdots + 1^p + \cdots + 0^p\right)^{1/p} = 1 \end{aligned}$$

Based on the above observation, our fitting problem consists in solving the following system of non-linear equations:

$$\begin{cases} (f_1(S_1)^p + \cdots + f_M(S_1)^p)^{1/p} = 1 \\ \cdots \\ (f_1(S_k)^p + \cdots + f_M(S_k)^p)^{1/p} = 1 \end{cases} \tag{5}$$

where k is the number of points in the front $\mathbb{F}_1$, $\forall f_i,\ z_i^{\min} = 0$ after the normalization, and p is the *unknown* (variable) of the equations. Unfortunately, there is no exact formula to solve an exponential equation of the form $a^p + b^p + \cdots = 1$ for any set of coefficients $(a, b, \ldots) \in \mathbb{R}^M$. Furthermore, numerical analysis methods (e.g., Newton's iterative method [8] or the LM algorithm [19]) are computationally expensive and not suitable for computing the value of p in each iteration with negligible overhead.

For the reasons above, we approximate the value of p using one single point from the front $\mathbb{F}_1$ for which the corresponding L_p exponential equation can be easily computed with an exact method. More specifically, we approximate p by considering the central point in the front $\mathbb{F}_1$. Since the objectives are normalized, the central point C is defined as the point in $\mathbb{F}_1$ with the minimum perpendicular distance to the vector $\vec{\beta}$ delimited by the ideal point $Z^{\min} = \vec{(0)}$ and nadir point $Z^{\max} = \vec{(1)}$:

$$C = \underset{f^n(S)}{\arg\min}\ dist^{\perp}(f^n(S), \vec{\beta}), \quad \forall S \in \mathbb{F}_1 \tag{6}$$

For example, let us consider the normalized front depicted in Figure 1. The extreme points A=(1,0) and B=(0,1) lie on the two objectives axes f_1^n and f_2^n while the central point C lies on the vector $\vec{\beta}$. Furthermore, the L_p norm of A and B is equal to one for any $p \in \mathbb{R}$. If we consider only the extreme points A and B and the central point C, our fitting problem consists in finding p such that the L_p norm of C is equal to the norm for A and B (which are both equal to one): $\|C\|_p = 1$. Since $\vec{\beta}$ is the bisector of the first quadrant, all coordinates of C are identical ($C_1 = C_2 = 0.5$). Therefore, the exponential equation to solve becomes $\left((0.5)^p + (0.5)^p\right)^{1/p} = 1$, whose exact solution is $p = 1$.

In general, given a generic central point C that lies on the bisector of the first quadrant ($\vec{\beta}$), the solution to the equation $\|C\|_p$ is:

$$\begin{aligned} \left(\sum_{i=1}^{M} C_i^p\right)^{1/p} = 1 \quad &\longrightarrow \quad M \cdot C_1^p = 1 \\ &\longrightarrow \quad C_1^p = 1/Mx \\ &\longrightarrow \quad p = -\frac{log(M)}{log(C_1)} \end{aligned} \tag{7}$$

where C_i denotes the i-th coordinate of the central point C.

In the equation above, we assumed that the C_i lies exactly on the bisector of the first quadrant. However, in practice, the central point of a generic non-dominated front $\mathbb{F}_1$ may have a distance $dist^{\perp}(C, \vec{\beta}) > 0$ and, thus, its coordinates in the objective space are not identical. For this reason, in AGE-MOEA (line 7 of Algorithm 1) we approximate the value of the exponent p using the equation:

$$p = -\frac{log(M)}{log\left(\frac{1}{M}\sum_{i=1}^{M} C_i\right)} = \frac{log(M)}{log(M) - log\left(\sum_{i=1}^{M} C_i\right)} \tag{8}$$

where M is the number of objectives; C is the central point in $\mathbb{F}_1$ and computed with Equation 6; and C_i is the i-th coordinate of C in the objective space. Since the coordinates of C cannot be all zero (otherwise C would coincide with the ideal point) nor all equal to ones (otherwise it would coincide with the nadir point), the summation $\sum_{i=1}^{M} C_i$ is always greater than zero and lower than M. This implies that the denominator in Equation 8 is always > 0.

If we use Equation 8 in the example of Figure 1, we obtain a value $p = -log(2)/(log(0.5 + 0.5) - log(2)) = 1$. The fitting unit curve for this example is flat and correspond to the line connecting the two extreme points A and B as shown in Figure 1. This straight line corresponds to the unitary curve for the norm L_1.

Complexity. The computational complexity to approximate the geometry of the first non-dominated front $\mathbb{F}_1$ (i.e.,the value of p) is $O(M \times N)$ where M is the number of objectives and N is the size of the population (in the worst case scenario $|\mathbb{F}_1| = N$). While the complexity for computing the formula in Equation 8 is $O(M)$.

3.4 Diversity, Proximity and Survival Scores

Once the geometry (Lp norm) of the first non-dominated front $\mathbb{F}_1$ is computed using Equation 8, we can measure both the diversity and proximity of $\mathbb{F}_1$ accordingly:

$$proximity(S) = \|f^n(S) - Z^{\min}\|_p = \|f^n(S)\|_p \tag{9}$$

$$diversity(S, \mathbb{F}_1) = \min_{T \neq S \in \mathbb{F}_1} \|f^n(S) - f(T)\|_p \tag{10}$$

The *proximity* score for a generic solution $S \in \mathbb{F}_1$ is measured as the distance (using the L_p norm) of its objective vector $f^n(S)$ to the ideal point. A solution $S \in \mathbb{F}_1$ that lies on the unit hypersurface associated with the estimated L_p norm has $proximity(S) = 1$ by definition (e.g.,

Algorithm 2: SURVIVAL-SCORE

Input:
$\mathbb{F}_d$: pool of non-dominated solutions
d: index of the non-dominated front
p: exponent of estimated geometry the p-norm

1 **begin**
2 **if** *d==1* **then**
3 score[E] $\longleftarrow +\infty$ /* E = extreme points of $\mathbb{F}_1$ */
4 $\Omega \longleftarrow$ E /* Considered solutions */
5 $\overline{\Omega} \longleftarrow \mathbb{F}_d \setminus \Omega$ /* Remaining solutions */
6 **for** *each solution* $S \in \overline{\Omega}$ **do**
7 proximity[S] $\longleftarrow \|f(S)\|_p$
8 **for** *each solution* $S_1 \in \mathbb{F}_d$ **do**
9 **for** *each solution* $S_2 \in \mathbb{F}_d$ **do**
10 dist[S_1, S_2] $\longleftarrow \|f(S_1) - f(S_2))\|_p$
11 **while** $|\overline{\Omega}| > 0$ **do**
12 **for** *each* $S \in \overline{\Omega}$ **do**
13 diversity[S] $\longleftarrow \min_{T \in \Omega} \text{dist}[S, T] + \min2_{T \in \Omega} \text{dist}[S, T]$
14 value[S] $\longleftarrow \dfrac{\text{diversity[S]}}{\text{proximity}[S]}$
  /* Select the solution with the max value */
15 $S^* \longleftarrow \arg\max_{S \in \overline{\Omega}} \text{value}[S]$
16 score[S^*] $\longleftarrow$ value[S^*]
17 $\Omega \longleftarrow \Omega \bigcup \{S^*\}$ /* Considered solutions */
18 $\overline{\Omega} \longleftarrow \overline{\Omega} \setminus \{S^*\}$ /* Remaining solutions */
19 **else**
20 **for** *each* $S \in \mathbb{F}_1$ **do**
21 score[S] $\longleftarrow 1/\|f(S)\|_p$

point C in Figure 1). A solution S with $proximity(S) < 1$ dominates parts of the unitary hypersurface of L_p (e.g., point D in Figure 1). Finally, a solution S with $proximity(S) > 1$ is more distant from the ideal point compared to the points in the unit hypersurface associated with L_p (e.g., point F in Figure 1). The diversity of the solutions $S \in \mathbb{F}_1$ is computed as the minimum distance (L_p norm) with the other solutions in the front $\mathbb{F}_1$.

The survival score of each solution $S \in \mathbb{F}_1$ combines both diversity (to maximize) and proximity (to minimize) as follows:

$$score(S) = \frac{diversity(S, \mathbb{F}_1)}{proximity(S)} \tag{11}$$

Algorithm 2 details to the procedure used in AGE-MOEA to assign survival scores to the solutions in each non-dominated front $\mathbb{F}_d$. The survival scores for the first non-dominated front $\mathbb{F}_1$ are computed in lines 3-18, while the score for the other non-dominated fronts $\mathbb{F}_{d>1}$ are computed in lines 20-21.

For what regards $\mathbb{F}_1$, first all extremes solutions are assigned the maximum possible survival score ($+\infty$) with the aim to preserve them in the population for the next generation (line 3 of Algorithm 2). Then, the procedure initializes two sets: (i) Ω that keeps track of all solutions with already assigned scores (line 4) and (ii) $\overline{\Omega}$ containing all solutions yet to score (line 5). In lines 6-7, the proximity scores for the solutions in $\overline{\Omega}$ are computed according to Equation 9, while the pairwise L_p distances between all solutions in $\mathbb{F}_1$ are computed in lines 8-10. Then, the survival score is computed within the loop in lines 12-14. In each loop, the procedure computes the diversity score for the solutions in $\overline{\Omega}$ considering the minimum (min) and the second minimum (min2) distances with regards to the solution in Ω (line 13 of Algorithm 2). In this way, the diversity of a solution S is computed with regards to solutions that have already been scored (or selected) in the previous iterations of the loop rather than considering all solutions in $\mathbb{F}_1$. A temporary survival score (value[S] in line 14) is then computed for each solution $S \in \overline{\Omega}$. The solution S^* with the maximum temporary score in $\overline{\Omega}$ is selected (line 15), and its final survival score is assigned in line 16. Then, the two sets Ω and $\overline{\Omega}$ are updated in lines 17-18. The temporary survival scores for the remaining solutions in $\overline{\Omega}$ are recomputed in the next iterations since we need to recompute their relative diversity with regards to the updated set Ω.

Finally, the survival scores for the solutions in the non-dominated fronts $\mathbb{F}_{d>1}$ are computed as the inverse of their proximity scores (lines 19-21 in Algorithm 2). Hence, dominated solutions closer to the unitary hypersurface induced by L_p have larger scores.

Complexity. The computational complexity of Algorithm 2 is $O(M \times N^2) + O(N^3)$, where M is the number of objectives and N is the population size. The elements of the overall complexity are:

- $O(M \times N)$ for computing the proximity scores in lines 6-7.
- $O(M \times N^2)$ for computing the distances for each pair of solutions in $\mathbb{F}_1$ (lines 8-10);
- $O(N^3)$ for the loop in lines 11-18. More specifically, the inner loop in line 12-14 has a complexity $O(|\overline{\Omega}| \times |\Omega|)$ and it is repeated $|\overline{\Omega}|$ (outer loop in line 11), where $\Omega \subseteq \mathbb{F}_1$ and $\overline{\Omega} \subseteq \mathbb{F}_1$.
- $O(M \times N)$ is the complexity for computing the survival score for the fronts $\mathbb{F}_{d>1}$.

4 EMPIRICAL STUDY

In our study, we considered the Maf test benchmark [2], with the number of objectives M=3, 5, and 10. The Maf suite has been designed by Cheng et al. [2] for the CEC 2017 competition on many objective optimization. The suite contains 15 test problems with different properties, such as linear (e.g., Maf1), concave (e.g., Maf2), convex (e.g., Maf3), multimodal (e.g., Maf4), degenerate (e.g., Maf6), disconnected (e.g., Maf7), and complex (e.g., Maf13) PFs. Therefore, such a suite is *a good representation of various real-world scenarios* [2].

To assess the effectiveness of AGE-MOEA, we compare its performance with five state-of-the-art many-objective evolutionary algorithms, namely NSGA-III [6], MOEA/D [12], GrEA [20], θ-DEA [22], and AR-MOEA [16]. These MOEAs use different strategies to balance proximity and diversity of the generated non-dominated fronts.

On each test problem, we run each algorithm 50 times to account for their randomness nature. In each independent run, we collected the non-dominated front produced by a given algorithm at the end of the search (i.e., when the maximum number of solution evaluations is reached) and computed the *inverted generational distance* (IGD) [23] to measure its overall quality. The IGD metric provides a single scalar value measuring both proximity to P^* and diversity of the solutions in the front P_A. The smaller the IGD

Table 1: Parameter values used for all many-objective evolutionary algorithm. n denotes the number of decision variables (length of the chromosomes).

Parameters	M=3	M=5	M=6
Population size [10] N	91	210	275
Number of Fitness Evaluations	27300	63000	82500
Number of iterations	300		
SBX probability [1]	$p_c = 1$		
SBX distributed index [1]	$\eta_c = 30$		
Polynomial mutation probability [5]	$p_m = 1/n$		
Mutation distributed index [5]	$\eta_m = 20$		

Table 2: Number of Maf problems [2] in which an algorithm A (e.g., AGE-MOEA) statistically outperforms (<) another algorithm B (e.g., NSGA-III) according to the Wilcoxon test (p-value≤0.05).

Comparison	M=3	M=5	M=10
AGE-MOEA < AR-MOEA	10	11	10
AR-MOEA < AGE-MOEA	2	2	3
AGE-MOEA < GrEA	12	12	12
GrEA < AGE-MOEA	2	2	0
AGE-MOEA < NSGA-III	13	14	15
NSGA-III < AGE-MOEA	2	0	0
AGE-MOEA < MOEA/D	12	14	12
MOEA/D < AGE-MOEA	3	1	2
AGE-MOEA < θ-DEA	13	13	14
θ-DEA < AGE-MOEA	1	0	1

values, the lower the distance between P_A and P^*, i.e., the better the performance of the algorithm A.

To assess the significance of the differences among the different MOEAs, we use the Wilcoxon rank-sum test [3] with significance level $\alpha = 0.05$. A significant p-value indicates that an algorithm A (e.g., AGE-MOEA) achieves significant lower IGD values than another algorithm B (e.g., NSGA-III) across 50 runs for a given test problem.

4.1 Implementation and Parameter settings

We implemented AGE-MOEA in MATLAB using PlatEMO, the open-source platform by Tian et al. [17] and public available on GitHub. The source code of AGE-MOEA (as well as all experimental results) will be made publicly available on GitHub after the double-blind review process. PlatEMO provides the source code for all benchmark problems as well as of the algorithms we use as baselines in our study. For all MOEAs, we used the same parameter setting reported in the related literature [1, 2, 5, 6, 10]. In MOEA/D, NSGA-III, and AR-MOEA, the population size N depends on the number of reference points generated with Das and Dennis's systematic approach [4, 6]. Instead, the population size can be arbitrary in AGE-MOEA since it does not use reference points. However, to ensure a fair comparison, we adopt the same population size and the same number of fitness evaluations 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.

Table 1 shows all parameters values for the evaluated MOEAs. For all the other parameters (e.g., the grid division in GrEA) we use the values suggested by their developers.

4.2 Empirical Results

In this subsection, we present the results of the comparison between AGE-MOEA and the five MOEAs baselines. Table 3 provides the average (mean) and the standard deviation (shown between parentheses) of the IGD achieved by each MOEA across 50 independent runs. The table also reports whether a baseline (e.g., MOEA) statistically outperforms (denoted with ↑), is statistically worst (denoted with ↓), or statistically equivalent (≈) to AGE-MOEA according to the Wilcoxon rank-sum test [3]. Besides, Table 2 summarizes the number of benchmark problems for which AGE-MOEA significantly outperforms a baseline MOEA (e.g., NSGA-III) and vice versa.

From Tables 3-2, AGE-MOEA performs significantly better than AR-MOEA in 10 out of 15 Maf test problems for M=3 and M=10. For M=5, AGE-MOEA performs statistically better than AR-MOEA in 11 out of 15 test problems. Particularly interesting is the Maf3 test problem, whose optimal Pareto front is convex (i.e., it is has a hyperbolic geometry with a norm $L_{p<1}$). Independently from the number of objectives, the IGD values achieved by AGE-MOEA are one or two orders of magnitude smaller than the IGD values achieved by AR-MOEA. For example, for M=10, AGE-MOEA obtains an average IGD value equal to $6.93\times10^{(}-2)$ compared to an IGD value of 1.55 achieved by AR-MOEA. On the other hand, AR-MOEA has significantly lower IGD values in 2 test problems (Maf6 and Maf11) for M=3, 2 test problems (Maf6 and Maf15) for M=5, and thee problems (Maf1, Maf9, and Maf15) for M=10. No significant difference is observed for the remaining few test problems.

For what regards the comparison with GrEA, we observe that AGE-MOEA statistically outperforms GrEA in 12 out of 15 test problems for M=3, 5, and 10. For some of the test problems, the differences between the two MOEAs are above one order of magnitude. For example, on Maf3 with M=10, AGE-MOEA obtains an average IGD value equal to 6.93×10^{-2} compared to an IGD value of $1.47\times10^{+4}$ achieved by GrEA (the difference is six orders of magnitude). In general, for M=10, there are 8 test problems out of 15 for which the IGD values of AGE-MOEA is at least one order of magnitude smaller (better) than those achieved by GrEA. Vice versa, GrEA is statistically better than AGE-MOEA in two test problems: Maf1 and Maf7 for M=3; Maf7 and Maf15 for M=5. Instead, for M=10 GrEA never achieves significantly better IGD values than AGE-MOEA. For the remaining test problems, there is no statistically significant difference between GrEA and AGE-MOEA.

From Tables 3-2, AGE-MOEA achieves significantly lower (better) IGD values than NSGA-III in 13, 14 and 15 out of 15 test problems for M=3, M=5, and M=10 respectively. The most significant difference is observed for Maf3 with M=10 where AGE-MOEA obtains an average IGD value equal to 6.93×10^{-2} while NSGA-III achieves an IGD value of $3.24\times10^{+3}$, which is five orders of magnitude larger than for AGE-MOEA. Vice versa, NSGA-III achieves statistically better IGD values in two test problems (Maf7 and Maf11) and only for M=3. Instead, NSGA-III never outperforms AGE-MOEA for M=5 and M=10. In particular, NSGA-III always produces significantly higher (worst) IGD values for all test problems when M is set to 10.

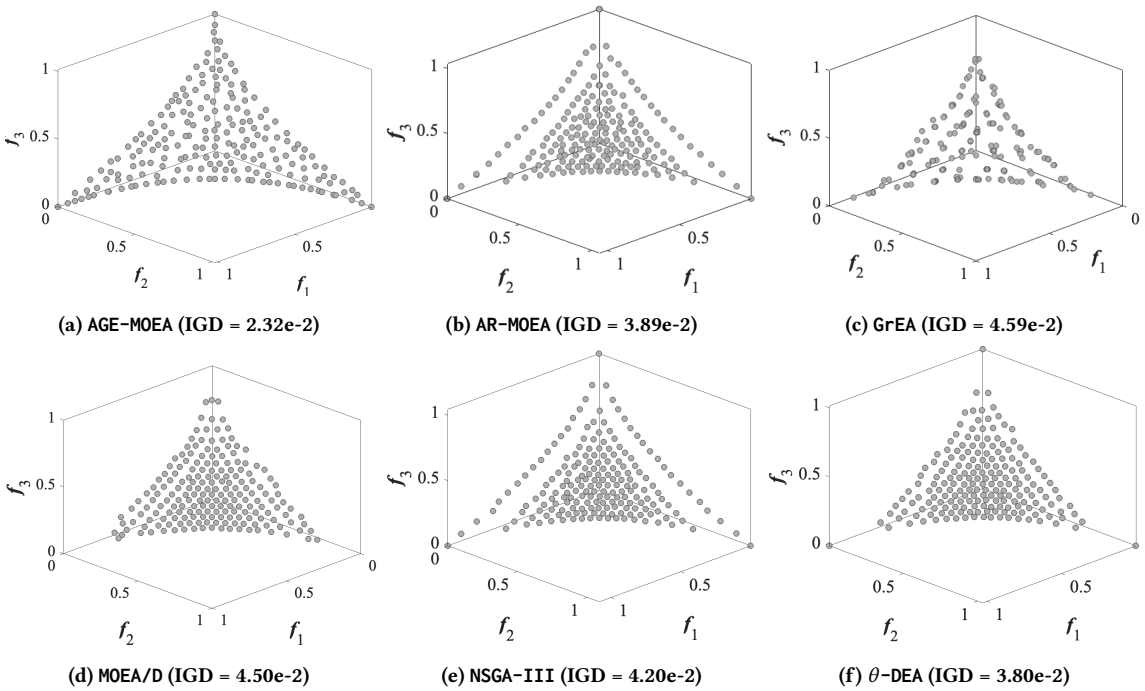
Regarding the comparison between AGE-MOEA and MOEA/D, we notice from Tables 3-2 that the former significantly outperforms the latter in the majority of test problems independently of the number of objectives. More specifically, AGE-MOEA achieves significantly

Table 3: IGD values (mean and standard deviation) achieved by the AGE-MOEA and the baselines on the Maf benchmark [2] with M=3,5, and 10 objectives. Best performance is highlighted in grey color.

Problem	M	AR-MOEA	GrEA	NSGA-III	MOEA/D	θ-DEA	AGE-MOEA
MaF1	3	4.3854e-2 (5.49e-4) ↓	4.2393e-2 (8.39e-4) ↑	6.1953e-2 (2.14e-3) ↓	7.0473e-2 (7.88e-6) ↓	8.0706e-2 (7.20e-4) ↓	4.3056e-2 (4.25e-4)
MaF2	3	3.2100e-2 (7.71e-4) ↓	3.1930e-2 (4.52e-4) ↓	3.6179e-2 (8.14e-4) ↓	4.1280e-2 (1.37e-3) ↓	3.6522e-2 (3.46e-4) ↓	3.1031e-2 (6.56e-4)
MaF3	3	1.5462e+0 (1.97e+0) ↓	9.9568e-1 (1.98e+0) ↓	2.3782e+0 (3.59e+0) ↓	3.1098e-1 (6.76e-1) ↑	3.3974e+0 (5.63e+0) ↓	5.1510e-1 (1.41e+0)
MaF4	3	1.2607e+0 (2.01e+0) ↓	1.3245e+0 (1.58e+0) ↓	3.1889e+0 (2.92e+0) ↓	2.1207e+0 (9.71e-1) ↓	1.4392e+0 (1.83e+0) ↓	7.5761e-1 (1.20e+0)
MaF5	3	1.0265e+0 (1.27e+0) ↓	9.3317e-1 (1.02e+0) ↓	7.1456e-1 (1.00e+0) ↓	1.2690e+0 (1.43e+0) ↓	8.1065e-1 (7.54e-1) ↓	3.0978e-1 (3.24e-1)
MaF6	3	5.1379e-3 (1.18e-4) ↑	2.0989e-2 (5.84e-4) ↓	1.4955e-2 (1.60e-3) ↓	7.9072e-2 (1.26e-1) ↓	3.3136e-2 (2.58e-3) ↓	5.4330e-3 (1.12e-4)
MaF7	3	1.9735e-1 (2.33e-1) ↓	8.6453e-2 (4.81e-3) ↑	7.8716e-2 (3.87e-3) ↑	1.7749e-1 (1.18e-1) ↓	1.0860e-1 (6.88e-2) ↓	9.1365e-2 (8.64e-2)
MaF8	3	9.5006e-2 (2.01e-2) ↓	8.6955e-2 (1.29e-2) ↓	1.3148e-1 (3.37e-2) ↓	2.9967e-1 (2.41e-1) ↓	1.9169e-1 (4.90e-2) ↓	7.8517e-2 (1.46e-2)
MaF9	3	8.1383e-2 (2.71e-2) ↓	4.6689e-1 (2.05e-1) ↓	8.9195e-2 (2.26e-2) ↓	1.3979e-1 (5.23e-2) ↓	8.4144e-2 (2.14e-2) ↓	6.8064e-2 (7.43e-3)
MaF10	3	3.3521e-1 (5.23e-2) ↓	2.4991e-1 (4.80e-2) ≈	4.3074e-1 (7.67e-2) ↓	5.7938e-1 (9.34e-2) ↓	3.9627e-1 (6.32e-2) ↓	2.2889e-1 (3.52e-2)
MaF11	3	1.6268e-1 (1.30e-3) ↑	2.3696e-1 (1.24e-2) ↓	1.6323e-1 (2.67e-3) ↑	2.6705e-1 (6.22e-2) ↓	1.5653e-1 (1.51e-3) ↑	1.7044e-1 (3.47e-3)
MaF12	3	2.2402e-1 (3.35e-3) ≈	2.5115e-1 (6.79e-3) ↓	2.3015e-1 (2.12e-2) ↓	2.9391e-1 (2.51e-2) ↓	2.2429e-1 (2.05e-3) ≈	2.2441e-1 (2.47e-3)
MaF13	3	9.0600e-2 (7.82e-3) ↓	1.7135e-1 (2.59e-2) ↓	9.4125e-2 (9.68e-3) ↓	1.1631e-1 (3.33e-2) ↓	9.2006e-2 (8.73e-3) ↓	7.9967e-2 (4.42e-3)
MaF14	3	9.8537e-1 (3.06e-1) ≈	1.3652e+0 (4.70e-1) ↓	1.2546e+0 (4.14e-1) ↓	6.1995e-1 (1.38e-1) ↑	1.2979e+0 (4.95e-1) ↓	9.3860e-1 (3.08e-1)
MaF15	3	3.8100e-1 (7.39e-2) ≈	5.8612e-1 (8.08e-2) ↓	7.2037e-1 (2.10e-1) ↓	3.7313e-1 (9.47e-2) ↑	9.0553e-1 (8.52e-2) ↓	4.2139e-1 (9.32e-2)
MaF1	5	1.1596e-1 (1.05e-3) ↓	1.2184e-1 (2.19e-3) ↓	1.8400e-1 (9.92e-3) ↓	1.5437e-1 (5.43e-2) ↓	2.0875e-1 (5.47e-3) ↓	1.1017e-1 (1.61e-3)
MaF2	5	9.4906e-2 (1.16e-3) ↓	1.0120e-1 (1.42e-3) ↓	1.1296e-1 (3.08e-3) ↓	1.1078e-1 (2.96e-4) ↓	1.2384e-1 (2.71e-3) ↓	9.2976e-2 (1.10e-3)
MaF3	5	1.5533e-1 (3.23e-1) ↓	1.9521e+0 (4.31e+0) ↓	1.3713e+0 (2.89e+0) ↓	1.0495e-1 (6.18e-3) ↓	9.3684e-2 (4.50e-3) ↓	5.4560e-2 (8.97e-4)
MaF4	5	2.2834e+0 (9.02e-2) ↓	2.3563e+0 (1.59e+0) ≈	3.0028e+0 (1.85e+0) ↓	1.1559e+1 (1.15e+0) ↓	2.8182e+0 (2.11e-1) ↓	1.8275e+0 (6.47e-2)
MaF5	5	1.9730e+0 (4.94e-3) ↓	1.9916e+0 (8.98e-1) ↓	2.0382e+0 (3.00e-1) ↓	7.5844e+0 (1.89e+0) ↓	1.9662e+0 (6.12e-3) ↓	1.7508e+0 (2.64e-2)
MaF6	5	2.0461e-3 (4.15e-5) ↑	3.5892e-2 (8.23e-4) ↓	1.6904e-2 (2.95e-3) ↓	1.1807e-1 (1.69e-1) ↓	8.1549e-2 (1.09e-2) ↓	2.4664e-3 (5.74e-5)
MaF7	5	2.5829e-1 (5.15e-3) ≈	2.3492e-1 (5.23e-3) ↑	2.8437e-1 (5.73e-3) ↓	5.1550e-1 (2.40e-2) ↓	3.0107e-1 (2.50e-2) ↓	2.5782e-1 (1.72e-2)
MaF8	5	8.9355e-2 (2.23e-3) ↓	1.2940e-1 (5.50e-3) ↓	1.6331e-1 (7.80e-3) ↓	2.5847e-1 (6.81e-2) ↓	3.0528e-1 (3.94e-2) ↓	7.6191e-2 (6.97e-4)
MaF9	5	8.8245e-2 (5.05e-3) ≈	1.1496e+0 (3.88e-1) ↓	3.5017e-1 (1.62e-1) ↓	1.3132e-1 (3.75e-2) ↓	6.7137e-1 (1.79e-1) ↓	8.8524e-2 (6.79e-3)
MaF10	5	6.8772e-1 (5.42e-2) ↓	5.1743e-1 (2.38e-2) ↓	8.2837e-1 (6.74e-2) ↓	8.7736e-1 (5.15e-2) ↓	6.4136e-1 (6.88e-2) ↓	4.1464e-1 (2.49e-2)
MaF11	5	3.9050e-1 (3.34e-3) ↓	4.9965e-1 (2.09e-2) ↓	3.8658e-1 (2.98e-3) ≈	7.7530e-1 (4.46e-2) ↓	3.8803e-1 (3.72e-3) ≈	3.8570e-1 (4.75e-3)
MaF12	5	9.4651e-1 (4.83e-3) ↓	9.3832e-1 (6.53e-3) ↓	9.3385e-1 (5.03e-3) ↓	1.6205e+0 (1.04e-1) ↓	9.2941e-1 (3.69e-3) ↓	9.2122e-1 (5.03e-3)
MaF13	5	9.5739e-2 (5.09e-3) ↓	3.8905e-1 (1.03e-1) ↓	1.7174e-1 (1.61e-2) ↓	1.6123e-1 (4.18e-2) ↓	2.9768e-1 (4.66e-2) ↓	8.9651e-2 (7.13e-3)
MaF14	5	8.1409e-1 (2.57e-1) ↓	7.5450e-1 (1.48e-1) ↓	8.4358e-1 (1.95e-1) ↓	7.8806e-1 (2.35e-1) ↓	1.0024e+0 (4.03e-1) ↓	6.5343e-1 (1.41e-1)
MaF15	5	5.6385e-1 (7.85e-2) ↑	7.3806e-1 (5.00e-2) ↑	1.2955e+0 (2.06e-1) ↓	6.3603e-1 (1.13e-1) ↑	1.0880e+0 (1.09e-1) ≈	1.0901e+0 (2.33e-1)
MaF1	10	2.2537e-1 (1.69e-3) ↑	2.3703e-1 (6.26e-3) ≈	2.8389e-1 (5.74e-3) ↓	5.3403e-1 (2.47e-2) ↓	3.1707e-1 (7.43e-3) ↓	2.3803e-1 (5.36e-3)
MaF2	10	1.8400e-1 (6.49e-3) ↓	3.6911e-1 (1.65e-2) ↓	2.0898e-1 (1.71e-2) ↓	2.6117e-1 (1.43e-3) ↓	1.9988e-1 (8.11e-3) ↓	1.7370e-1 (5.34e-3)
MaF3	10	1.5649e+0 (5.04e+0) ↓	1.4697e+4 (1.69e+4) ↓	3.2428e+3 (6.34e+3) ↓	1.4320e-1 (2.11e-3) ↓	8.5241e+0 (2.40e+1) ↓	6.9331e-2 (1.01e-3)
MaF4	10	9.5813e+1 (6.24e+0) ↓	1.6292e+2 (3.91e+1) ↓	9.4722e+1 (8.70e+0) ↓	5.3804e+2 (4.52e+1) ↓	1.1635e+2 (1.39e+1) ↓	5.7991e+1 (4.15e+0)
MaF5	10	9.7353e+1 (5.14e+0) ↓	4.7038e+1 (1.05e+0) ≈	7.7391e+1 (1.23e+0) ↓	3.0165e+2 (2.68e+0) ↓	7.7560e+1 (7.13e-1) ↓	4.6762e+1 (1.57e+0)
MaF6	10	3.4000e-1 (3.17e-1) ≈	8.6533e-1 (4.10e-1) ↓	6.1137e-1 (2.11e-1) ↓	1.1737e-1 (1.62e-1) ↑	1.6267e-1 (2.56e-1) ↑	3.5746e-1 (1.78e-1)
MaF7	10	1.4115e+0 (8.58e-2) ↓	2.6086e+0 (6.41e-2) ↓	1.1407e+0 (7.72e-2) ↓	2.4215e+0 (5.00e-1) ↓	9.0120e-1 (4.96e-2) ↓	8.4737e-1 (9.93e-3)
MaF8	10	1.2401e-1 (3.31e-3) ↓	1.4432e-1 (4.02e-3) ↓	2.7530e-1 (4.98e-2) ↓	9.6382e-1 (2.25e-2) ↓	7.5206e-1 (9.76e-2) ↓	1.0316e-1 (6.42e-4)
MaF9	10	1.5808e-1 (7.75e-3) ↑	1.4224e+0 (5.86e-2) ↓	4.4905e-1 (1.09e-1) ↓	1.3035e+0 (1.61e+0) ↓	7.8762e-1 (1.46e-1) ↓	2.9156e-1 (9.70e-3)
MaF10	10	1.5107e+0 (1.01e-1) ↓	1.1179e+0 (4.06e-2) ↓	1.6448e+0 (1.35e-1) ↓	1.9093e+0 (1.39e-1) ↓	1.1739e+0 (8.54e-2) ↓	9.2879e-1 (1.74e-2)
MaF11	10	9.9968e-1 (3.65e-2) ↓	1.1312e+0 (3.55e-2) ↓	1.2069e+0 (1.97e-1) ↓	1.9329e+0 (4.15e-2) ↓	1.1605e+0 (1.44e-1) ↓	9.7470e-1 (1.28e-2)
MaF12	10	4.4564e+0 (2.82e-2) ↓	4.1458e+0 (4.01e-2) ↓	4.3421e+0 (7.55e-2) ↓	8.9572e+0 (1.64e-1) ↓	4.3330e+0 (3.25e-2) ↓	3.9226e+0 (2.99e-2)
MaF13	10	1.1361e-1 (5.45e-3) ↓	4.4202e-1 (2.39e-1) ↓	2.2892e-1 (3.43e-2) ↓	9.4977e-1 (6.99e-2) ↓	5.6955e-1 (9.15e-2) ↓	9.6785e-2 (9.11e-3)
MaF14	10	6.2926e-1 (4.32e-2) ≈	1.1096e+0 (2.47e-1) ↓	1.2089e+0 (3.12e-1) ↓	4.6465e-1 (1.04e-1) ↑	1.0947e+0 (4.30e-1) ↓	6.1479e-1 (3.64e-2)
MaF15	10	8.9098e-1 (1.31e-1) ↑	1.3888e+0 (4.96e-1) ≈	1.3848e+0 (2.28e-1) ↓	1.0341e+0 (9.57e-2) ≈	1.2753e+0 (1.41e-1) ↓	1.0738e+0 (1.46e-1)

better IGD values in 12 problems (out of 15) for M=3, in 14 problems form M=5, and in 12 problems for M=10. The largest difference between the two MOEAs is observed for the Maf6 test problem with M=5. The optimal Pareto front for Maf6 is concave (i.e., it has a spherical geometry with $L_{p>1}$ norm) a degenerate [2]. In this case, AGE-MOEA has an average IGD value equal to 2.46×10^{-3} while MOEA/D has an IGD value of 1.18×10^{-1} (the difference is six orders of magnitude). MOEA/D significantly outperforms AGE-MOEA in three problems (Maf3, Maf14, and Maf15) for M=3, in one problem (Maf15) for M=5, and two problems (Maf6 and Maf14) for M=5.

Concerning θ-DEA, we observe that AGE-MOEA outperforms θ-DEA in 13 out of 15 test problems for M=3 and M=15; and in 14 out of 15 test problems for M=10. Independently of the number of objectives M, there are six test problems for which the IGD values achieved by AGE-MOEA are at least one order of magnitude smaller than the values achieved by θ-DEA On the other hand, θ-DEA performs significantly better than AGE-MOEA in only one test problem for M=3 (Maf11) and M=10 (Maf6). For Maf12 with M=3 and for Maf11 and Maf15 with M=10, we do not observe any statistically significant difference between GrEA and AGE-MOEA.



(a) AGE-MOEA (IGD = 2.32e-2) (b) AR-MOEA (IGD = 3.89e-2) (c) GrEA (IGD = 4.59e-2)

(d) MOEA/D (IGD = 4.50e-2) (e) NSGA-III (IGD = 4.20e-2) (f) θ-DEA (IGD = 3.80e-2)

Figure 2: Fronts produced by the different MOEAs for Maf3 with M=3 and population size N=190.

Based on the results above, we can conclude that AGE-MOEA *has the best overall performance (measured using the IGD metrics) compared to five stat-of-the=art MOEAs experimented in this study.*

Graphical comparison. To provide a graphical comparison between the different MOEAs experimented in this study, Figure 2 depicts the Pareto front achieved for Maf3 with M=3. For the sake of this analysis, we increased the population size to N=190, which is the number of reference points obtained with the Das and Dennis's systematic approach [4, 6]. For the other parameters, we used the same setting described in Table 1. To have a more reliable comparison, we ran each MOEA 51 time and selected the Pareto front corresponding to the run with the median IGD score. We considered Maf3 because its Pareto optimal front forms a hyperbolic surface (hyperbolic geometry with L_p norm having $p < 1$) that intersect the objective axes in the points $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$.

As we can observe, AGE-MOEA produced a set of non-dominated solutions that are well-distributed and that cover the entire hyperbolic surface. The corresponding median IGD value is 2.32×10^{-2}. Unlike AGE-MOEA, the other five state-of-the-art MOEAs generate fronts with uneven data points distributions. In particular, the majority of the generated solutions are located in the central region of the hyperbolic surface. This uneven distribution is due to the fact that the majority of the reference points (e.g., in NSGA-III) intersect a hyperbolic surface (L_p norm with $p < 1$) in the central region of the surface itself. The stronger the curvature of the hyperbolic surface $\mathbb{S}_h$ (i.e., the lower the exponent $0 < p < 1$), the smaller the region in the center of the surface in which most of the reference points are located.

AR-MOEA uses a reference point adaptation method to adjust the reference points at each generation for the calculation of the indicator IGD-NS [16]. However, based on our simulation with Maf3, it produces a set of reference points that covers the central region at the expense of the solutions that are closer to the border (corner) of the surface.

5 CONCLUSION AND FUTURE WORK

In this paper, we have introduced a novel MOEA, namely AGE-MOEA, whose selection procedure considers both the *diversity* among the population members, and the *proximity* of each population member to the ideal point. Unlike state-of-the-art MOEAs, AGE-MOEA does not make any assumption about the geometry of the Pareto front. Instead, it estimates the geometry of the front by using a fast procedure, whose computational complexity is lower than the complexity of the non-dominated sorting algorithm.

We assessed the performance of AGE-MOEA using the Maf test suite, which comprises 15 many-objective problems, with different characteristics, and varying the number of objectives $M \in \{3, 5, 10\}$. Besides, we compared the performance of AGE-MOEA with five state-of-the-art many-objectives evolutionary algorithm, namely NSGA-III, MOEA/D, GrEA, θ-DEA, and AR-MOEA. The achieved results show that AGE-MOEA significantly outperforms the five baselines independently of the number of objectives.

As future work, we plan to investigate alternative heuristics to estimate the geometry of the non-dominated fronts. We also plan to apply AGE-MOEA on constrained MOPs, real-world problems, and more test benchmarks.

REFERENCES

[1] Ram Bhushan Agrawal, K Deb, and RB Agrawal. 1995. Simulated binary crossover for continuous search space. *Complex systems* 9, 2 (1995), 115–148.

[2] Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. 2017. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems* 3, 1 (01 Mar 2017), 67–81. https://doi.org/10.1007/s40747-017-0039-7

[3] W. J. Conover. 1998. *Practical Nonparametric Statistics* (3rd edition ed.). Wiley.

[4] 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.

[5] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.

[6] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 577–601. https://doi.org/10.1109/TEVC.2013.2281535

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

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

[9] Xiaoyu He, Yuren Zhou, Zefeng Chen, and Qingfu Zhang. 2018. Evolutionary Many-objective Optimization based on Dynamical Decomposition. *IEEE Transactions on Evolutionary Computation* (2018).

[10] H. Jain and K. Deb. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 602–622. https://doi.org/10.1109/TEVC.2013.2281534

[11] Saúl Zapotecas Martínez, Víctor A Sosa Hernández, Hernán Aguirre, Kiyoshi Tanaka, and Carlos A Coello Coello. 2014. Using a family of curves to approximate the Pareto front of a multi-objective optimization problem. In *International Conference on Parallel Problem Solving from Nature*. Springer, 682–691.

[12] Suat Özdemir, A Attea Bara'a, and Önder A Khalil. 2013. Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks. *Wireless personal communications* 71, 1 (2013), 195–215.

[13] Robin C Purshouse and Peter J Fleming. 2007. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 770–784.

[14] Ryoji Tanabe and Akira Oyama. 2017. Benchmarking MOEAs for multi-and many-objective optimization using an unbounded external archive. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 633–640.

[15] Anthony C Thompson and Anthony C Thompson. 1996. *Minkowski geometry*. Vol. 63. Cambridge University Press.

[16] Ye Tian, Ran Cheng, Xingyi Zhang, Fan Cheng, and Yaochu Jin. 2018. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation* 22, 4 (2018), 609–622.

[17] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Computational Intelligence Magazine* 12 (2017), 73–87. Issue 4.

[18] Ye Tian, Xiaoshu Xiang, Xingyi Zhang, Ran Cheng, and Yaochu Jin. 2018. Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–6.

[19] Ye Tian, Xingyi Zhang, Ran Cheng, Cheng He, and Yaochu Jin. 2018. Guiding Evolutionary Multio-bjective Optimization With Generic Front Modeling. *IEEE transactions on cybernetics* (2018).

[20] Shengxiang Yang, Miqing Li, Xiaohui Liu, and Jinhua Zheng. 2013. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 17, 5 (2013), 721–736.

[21] Yuan Yuan, Hua Xu, and Bo Wang. 2014. An improved NSGA-III procedure for evolutionary many-objective optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 661–668.

[22] Yuan Yuan, Hua Xu, Bo Wang, and Xin Yao. 2016. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 16–37.

[23] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.