Using Archiving Methods to Control Convergence and Diversity for Many-Objective Problems in Particle Swarm Optimization

Andre Britto  
Federal University of Parana  
Curitiba, Parana, Brazil 81531-980  
andreb@inf.ufpr.br

Aurora Pozo  
Federal University of Parana  
Curitiba, Parana, Brazil 81531-980  
aurora@inf.ufpr.br

Abstract—Multi-Objective Particle Swarm Optimization (MOPSO) is a population based multi-objective meta-heuristic inspired on animal swarm intelligence. It is used to solve several Multi-Objective Optimization Problems (MOPs), problems with more than one objective function. However, Multi-Objective Evolutionary Algorithms (MOEAs), including MOPSO, have some limitations when the number of objective grows. Many-Objective Optimization research methods to decrease the negative effect of applying MOEAs into problems with more than three objective functions, In this context, the goal of this work is to explore several archiving methods from the literature used by MOPSO to store the selected leaders into Many-Objective Problems. Moreover, new archiving methods are proposed specially for these problems. The use of the archiving methods into MOPSO is evaluated through an empirical analysis aiming to observe the impact of these methods in the convergence and the diversity to the Pareto front, in Many-Objective scenarios.

I. INTRODUCTION

Multi-objective optimization problems (MOPs) involve the simultaneously optimization of two or more conflicting objectives subject to certain constraints. Evolutionary algorithms, including Multi-Objective Particle Swarm Optimization (MOPSO), have demonstrated to be very powerful, dealing with MOPs in a suitable way and providing a set of good solutions for the problem considering Pareto non-dominance concepts [1].

Particle Swarm Optimization (PSO) [2] is a population based meta-heuristic inspired on animal swarm intelligence. PSO has been widely used to solve many optimization problems. In PSO, a swarm is composed by particles which use simple local rules to govern their actions and by means of the interactions of the entire group, through the definitions of leaders (best solutions), the swarm achieves its objectives. Since in MOPs there is no a single best solution, in MOPSO there is a set of leaders, stored in an external archive that must guide the swarm to the best areas of the search space. Therefore, two main issues for MOPSO to deal with MOPs are: how to select the leaders and how to store these leaders in the external archive.

These leaders are stored into the external archive through archiving methods [3]. Basically, given a new objective vector generated by any Multi-Objective Evolutionary Algorithm (MOEA) the goal of an archiver is to maintain the best solutions (only non-dominated solutions) found so far in the search [1]. This archive typically is limited by a size $N$ and the archiver must use different strategies to introduce aspects like convergence and diversity to the search. Archiving methods were previously explored in [3]. There, it is presented a work that studies the behavior of different archivers in MOPs and through a theoretical analysis it is showed the influences of each archiver in the search.

In spite of the good results of MOEAs, including MOPSO algorithms [4], in recent years, researchers have focused on the work with a larger number of objectives. This occurs because many real world problems have more than three objectives and experimental studies have shown that MOEAs scale poorly in this context [5], [6]. Problems with more than three objectives are called Many-Objective Optimization Problems (MaOPs). One of the main challenges faced by MOEAs when applied to MaOPs is the deterioration of the search ability. This deterioration occurs due to the increase of the number of non-dominated solutions when the number of objectives grows and, consequently, there is no pressure towards the Pareto front. The research in this field aim to overcome these limitations. In these works, the MOPSO is a topic little explored, but it has important features like the selection of the leaders and the archiving methods that can be explored to obtain good results in MaOPs. Ibanez et al. studied the behavior of archivers in MOPs in [3], but this work only deals with a low number of objectives.

In the literature MaOPs are usually solved by: the definition of new preference relation, the dimensionality reduction or decomposition strategies [7]. Our main goal is to explore some aspects of MOPSO and enhance them to obtain a better performance in MaOPs. In this work, different archiving methods from the literature are explored and theirs influence in MOPSO when the number of objective grows is analyzed. Besides, seeking to obtain better results, here, some new archiving methods are proposed. The proposed methods are: Ideal Archiver, Distributed Archiver and Distance to Reference Points Archiver. The idea of these methods is to refine the...
search through the selection of some preference regions that will guide the archiving process, specially for MaOPs.

An empirical analysis is performed to evaluate all archiving methods in MaOPs. SMSPO algorithm is used as basis algorithm. [4]. It is used the DTLZ2 many-objective benchmarking problem [8]. This problem was chosen to emphasize the study of the convergence to the Pareto Front. A set of quality indicators is used to investigate how these archiving methods affect convergence and diversity of MOPSO search in many objective scenarios, they are: Generational Distance (GD), Inverse Generational Distance (IGD), Spacing, Hypervolume and also it is analyzed the distribution of the Tchebycheff distance over the "knee" of the Pareto front [9].

The rest of this paper is organized as follows: Section II describes main concepts of Many-Objective Optimization. The Particle Swarm Optimization is presented in Section III. Section IV analyzes the archiving method from the literature and describes the proposed archiving methods. Section V presents the empirical analysis performed to evaluate the archiving methods, the methodology used in the experiments and the discussion of results. Finally, Section VII presents the conclusions and future works.

II. MANY-OBJECTIVE OPTIMIZATION

A Multi-Objective problem (MOP) involves the simultaneous satisfaction of two or more objective functions. Furthermore, in such problems, the objectives to be optimized are usually in conflict, which means that they do not have a single best solution, but a set of solutions. To find this set of solutions it is used the Pareto Optimality Theory [1]. The general multi-objective minimization problem, without constraints, can be stated as (1).

\[
\text{Minimize } f(x) = (f_1(x), f_2(x), \ldots, f_m(x))
\]

subject to \(x \in \Omega\), where: \(x \in \Omega\) is a feasible solution vector, \(\Omega\) is the feasible region of the problem, \(m\) is the number of objectives and \(f_i(x)\) is the \(i\)-th objective function of the problem.

In this case, the purpose is to optimize \(m\) objective functions simultaneously, with the goal to find a good trade-off of solutions that represent the better compromise between the objectives. So, in MOP the optimum is defined through terms of Pareto Optimality, Pareto Dominance, Pareto Optimal Set and Pareto Front [1].

**Definition 1, Pareto Dominance:** Given \(f(x) = (f_1(x), f_2(x), \ldots, f_m(x))\) and \(f(y) = (f_1(y), f_2(y), \ldots, f_m(y))\), \(f(x)\) dominates \(f(y)\), denoted by \(f(x) \prec f(y)\), if and only if (minimization):

\[
\forall i \in \{1, 2, \ldots, m\} : f_i(x) \leq f_i(y) \text{, and }
\exists i \in \{1, 2, \ldots, m\} : f_i(x) < f_i(y)
\]

\(f(x)\) is non-dominated if there is no \(f(y)\) that dominates \(f(x)\).

**Definition 2, Pareto Optimality:** Given a solution \(x \in \Omega\), with objective vector \(f(x) = (f_1(x), f_2(x), \ldots, f_m(x))\), \(x\) is said Pareto Optimal if and only if there is no \(y \in \Omega\), with objective vector \(f(y) = (f_1(y), f_2(y), \ldots, f_m(y))\), such as \(f(y)\) dominates \(f(x)\).

**Definition 3, Pareto Optimal Set:** For a MOP, the Pareto Optimal Set, \(P^*\), is the set of the best solutions \(x \in \Omega\) (in the decision variable space). \(P^*\) is defined as follows:

\[
P^* := \{x \in \Omega \mid \nexists y \in \Omega , f(y) < f(x)\}
\]

**Definition 4, Pareto Front:** Each solution in \(P^*\) has a non-dominated image in the objective space, called Pareto Front. Given a Pareto Optimal Set, \(P^*\), for some MOP, the Pareto Front, denoted by \(PF^*\) correspond to the set of objective vectors of each solution \(x \in P^*\). \(PF^*\) is defined by:

\[
PF^* := \{f(x) \mid x \in P^*\}
\]

In MOP, MOEAs modify Evolutionary Algorithms (EAs) [1] in two ways: they incorporate a selection mechanism based on Pareto optimality and they adopt a diversity preservation mechanism that avoids the convergence to a single solution. The purpose is to discover solutions that are not dominated by any other in the objective space. In most applications, the search for the Pareto optimal set is NP-hard [1], so the MOEAs focuses on finding an approximation set, as close as possible to the Pareto Front. MOEAs are particularly suitable for this task because they simultaneously evolve a population of potential solutions for the problem by obtaining a set of solutions to approximate the Pareto front in a single execution of the algorithm.

In spite of most of the studies on MOPs have focused on problems with a few numbers of objectives, practical optimization problems involve a large number of criteria. Therefore, research efforts have been oriented to investigate the scalability of these algorithms with respect to the number of objectives [5]. MOPs having more than 3 objectives are referred as Many-Objective Optimization Problems, MaOPs, and the field that studies new solutions for these problems is called Many-Objective Optimization.

Several studies have showed that MOEAs scale poor in many-objective optimization problems [5] [9] [10] [11]. The main reason for this is that the number of non-dominated solutions increases exponentially with the number of objectives. As consequence: first, the search ability is deteriorated because it is not possible to impose preferences for selection purposes; Second, the number of solutions required for approximating the entire Pareto front also increases, and finally there exists difficulty of the visualization of solutions.

To avoid these problems, currently, the research community has tackle these issues using mainly three approaches [5]: the adaptation of preference relations that induce a finer order on the objective space [9]; the dimensionality reduction, where the overall idea is to identify the least non conflicting objectives and discard them; and decomposition strategies that uses decomposition methods, which have been studied in the mathematical programming community, into evolutionary algorithms for multi-objective optimization.
Our main goal, is to study the effectiveness of PSO algorithm in Many-Objective Problems. Until fairly recently most of the research was concentrated on a small group of algorithms, often Genetic Algorithms. In the literature some works deal with MaOPs using PSO algorithms. Here, it is explored specific features of PSO algorithm and we try to enhance them to deal with MaOPs. So, instead of changing the dominance relation, we will investigate the importance of archiving methods in MOPSO algorithms, for MaOPs. Next section will discuss some aspects of MOPSO.

III. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a cooperative population-based heuristic inspired by the social behavior of birds flocking to find food [2]. In Particle Swarm Optimization the system initializes a set of solutions and searches for optimal solutions by updating generations. The set of possible solutions is a set of particles, called a swarm, which moves in the search space, in a cooperative search procedure. These movements are performed by the velocity operator that is guided by a local and a social component. To extend PSO for multi-objective optimization, Multi-Objective Particle Swarm Optimization (MOPSO), it is necessary to modify the mechanism of guidance of the algorithm. The Pareto dominance relation is adopted to establish preferences among solutions to be considered as leaders. By exploring the Pareto dominance concepts, each particle in the swarm could have different leaders, but only one may be selected to update the velocity.

This set of leaders is stored in an external archive that contains the best non-dominated solutions found so far. Often, the archive of leaders has a maximum size. When this archive gets full, the algorithm has to decide which particles must remain in the external archive for the next iteration. In literature there are different archiving methods to make this decision. Here, some archiving methods will be studied, and will be discussed how these methods influence MOPSO in many-objective scenarios. The archiving methods will be discussed in Section IV.

The basic steps of a MOPSO algorithm are: initialization of the particles, computation of the velocity, position update and update of leader’s archive.

Each particle $p_i$, at a time step $t$, has a position $x(t) \in \mathbb{R}^n$, that represents a possible solution. The position of the particle, at time $t + 1$, is obtained by adding its velocity, $v(t) \in \mathbb{R}^n$, to $x(t)$, Equation 4. The velocity of a particle $p_i$ is based on the best position already fetched by the particle, $\overrightarrow{p}_{best}(t)$, and the best position already fetched by the set of neighbors of $p_i$, $\overrightarrow{R}_h(t)$, that is a leader from the repository, see Equation 5.

$$\overrightarrow{x}(t + 1) = \overrightarrow{x}(t) + \overrightarrow{v}(t + 1) \quad (4)$$

$$\overrightarrow{v}(t + 1) = \varpi \cdot \overrightarrow{v}(t) + (C_1 \cdot \phi_1) \cdot (\overrightarrow{p}_{best}(t) - \overrightarrow{x}(t))$$

$$+ (C_2 \cdot \phi_2) \cdot (\overrightarrow{R}_h(t) - \overrightarrow{x}(t)) \quad (5)$$

The variables $\phi_1$ and $\phi_2$, in (5), are coefficients that determine the influence of the particle best position. Constants $C_1$ and $C_2$ indicate how much each component influences on velocity. The coefficient $\varpi$ is the inertia of the particle, and controls how much the previous velocity affects the current one. $\overrightarrow{R}_h$ is a particle from the repository, chosen as a guide of $p_i$.

In literature there are several MOPSO algorithm proposed. In this work, the SMPSO algorithm is used as basis. The SMPSO algorithm was presented in [4]. This algorithm limits the velocity of the particles. In this algorithm the velocity of the particle is limited by a constriction factor $\chi$, that varies based on the values of $C_1$ and $C_2$. Besides, the SMPSO introduces a mechanism that links or constraints the accumulated velocity of each variable $j$ (in each particle). The equations of SMPSO are presented in [4]. After the velocity of each particle has been updated a mutation operation is applied. A polynomial mutation is applied [12] in 15% of the population, randomly selected. The global leader, $\overrightarrow{R}_h$, is chosen by a binary tournament. The archiving method uses Crowding Distance [12] and will be discussed in next section.

IV. ARCHIVING METHODS

The idea of an external archive to maintain the best solutions found so far in the search is used in most of the MOEAs [1]. A formalization of a basic archiver with size $N$ and a simple class of archivers called precise archivers are presented in [13]. Simply, given a sequence of points (objective vectors generated by a MOEA) the goal of an archiver is to maintain a subset of these points in an archive $A$, where $|A| \leq N$. Also, $A'$ is denoted the archive after the presentation of the $t-th$ objective vector [3].

Given an archive $A'$, it only contains non-dominated objective vectors (and its respective coordinates in the search space) [13]. A precise archiver $A$, where $|A| \leq N$, presented at Algorithm 1, has some properties: (1) given a solution $x$, if $x$ dominates some solution in $A^{t-1}$, $x$ is added in $A^{t-1}$ and the dominated points are removed, obtaining $A'$; (2) If $x$ is dominated by any point in $A^{t-1}$, $A'$ remains unchanged then $A^{t-1} = A'$; (3) If $|A^{t-1}| < N$, $x$ does not dominate any solution in $A^{t-1}$ and is not dominated by any solution in $A^{t-1}$, $x$ is added to $A'$; (4) If the archive is full, $x$ does not dominate and is not dominated by any solution in $A^{t-1}$, a function filter() is applied to define which solutions will remain in $A^{t-1} \cup x$.

In [3], it is presented an important work that analyzes the behavior of several archiving methods in MOPs. In this work, eight archivers are studied and important features of the archivers are observed through a theoretical and experimental analysis. However, although the results presented in [3] indicate important distinguishing features of different archivers of the literature, these methods still are not explored in MaOPs. Our goal, is to explore how an archiver influences the search of a MOPSO algorithm when the number of objective grows.

In this study, it is used 12 different archivers, 10 follow the precise archiver scheme, only those based on $\epsilon$-dominance [14] follow a different scheme. 7 of the 12 archivers are obtained from the literature and were previously
Algorithm 1 Precise Archive

Input: $A^{t-1}$, $x$, $N$ (Previous archive, new solution, Maximum size of the archive)

Output: $A^t$

1. If $\exists y \in A^{t-1}, x < y$ then
2. $A^t \leftarrow A^{t-1} \cup x/\forall y \in A^{t-1}, x < y$ (1)
3. else If $\exists y \in A^{t-1}, y < x$ then
4. $A^t \leftarrow A^{t-1}$ (2)
5. else If $\nexists y \in A^{t-1}, x < y$ or $y < x$
6. If $(|A| < N)$
7. $A^t \leftarrow A^{t-1} \cup x$ (3)
8. else $A^t \leftarrow \text{filter}(A^{t-1}, x)$ (4)

Fig. 1: Precise Archiver pseudo-code. The numbers in bold represent each properties of the precise archiver.

analyzed in [3]. 4 are proposed here and the last one is a random archiver. Following, the archiving methods from literature are described and at next section the proposed methods are presented. The names between brackets indicate the abbreviations used in the experiments.

Adaptive Grid archiver (AG): The Adaptive Grid was proposed in [15] and has the idea to enforce an uniform distribution of non-dominated solutions by placing these solutions in different cells of a grid. In this grid, the objective space is divided in order to measure the density of different regions. The Adaptive Grid archiver uses this grid as a mechanism to introduce more diversity on the selected solutions. In this archiver, the stored solutions are organized in the grid and the \text{filter}(A^{t-1}, x) function adds $x$ to the archive and then remove one random solution from the most crowded cell.

Crowding Distance, NSGA2 archiver (CD) (Crowd): This archiver uses the Crowding Distance (CD) [12] measure to decided which solutions still remain in the external archive.

This archiver introduces more diversity on the search and the \text{filter}(A^{t-1}, x) function adds $x$ to the archive and removes the solutions with worst CD, i.e., the solution placed on a crowded region in the objective space. This archiver is used by the SMPSO, the basis algorithm in this work.

Dominating archiver (Dom): This is a simple archiver, that only adds to the archive solutions that have dominated any other solution in the archive. Basically, \text{filter}(A^{t-1}, x) returns $A^{t-1}$.

Adaptive $\epsilon$-Approx Archiving (eapp): This archiver uses concepts of $\epsilon$-dominance [14]. Basically, instead of using the Pareto dominance, this archiver uses the $\epsilon$-dominance. So, it only accepts one new solution if it $\epsilon$-dominates any other in the external archive [14]. Also, in [14] is presented an adaptive scheme that allows an user to specify a size $N$, rather than the $\epsilon$ value. However, here we follow the approach presented in [3], where the precise archive scheme is not followed and the archiver does not have an user-defined maximum size $N$. Instead, the $\epsilon$ value is defined as an user parameter.

Adaptive $\epsilon$-Pareto Archiving (eaps): This archiver discretizes the objective space into different boxes defined by the value of $\epsilon$ [14]. The basic idea is that every non-dominated point belongs to a specific box. If two or more non-dominated point are placed into the same box, only the point closer to the box origin remain as a non-dominated solution. In the same way of the previous archiver, this archiver does not follow the precise archive scheme and does not have a fixed size $N$.

Multi-level Grid Archiving (MGA): This archiver, proposed in [16], has the idea to combine the Adaptive Grid scheme with the $\epsilon$-Pareto Archiving. This archiver follows the scheme of the precise archiver and is limited by maximum size $N$. If the archive is full, the \text{filter}(A^{t-1}, x) function divides the objective space into boxes and every point in the archive has a box index. After, it is observed the dominance relation between the box index. If the new solution to be added, $x$, belongs to one of the dominated boxes, it is removed, otherwise one of the points that have dominated boxes are removed randomly [16]. If there is no dominance relation between the boxes, the objective space is re-divided into smaller boxes until a dominated box is found.

Random archiver (rand): The function \text{filter}(A^{t-1}, x) removes a random solution from $A^{t-1} \cup x$.

Unbound archiver (ub): This archiver does not have a maximum size $N$. Every non-dominated solution is added and the \text{filter}(A^{t-1}, x) returns $A^{t-1} \cup x$.

A. Proposed archiving methods

Besides the archiving methods from the literature, here three new archiving methods are proposed, specially designed to explore characteristics like convergence and diversity. All approaches follow the precise archive scheme. The proposed methods directly explore two different ideas: to spread the solutions in the archive to different areas of the objective space obtaining more diversity or to concentrate the solutions from the archive in specific areas of the objective space introducing more convergence in the search. Following, the proposed approaches are presented.

Ideal archiver (ideal): The first approach has the idea to guide the solutions in the archive to a specific area of the objective space, with the goal to increase the convergence of the search. So, in the proposed method the ideal point [1] is selected as guide. The ideal point is a vector with the best value for every objective value, obtained at each iteration between the points in the external archive. In this approach, the distance to the ideal point is used to define which solutions will remain in the archive. The distance to the ideal point was previously applied into Many-Objective Optimization in [17], but it was used to define a new preference relation. Here, first the function \text{filter}(A^{t-1}, x) obtains the ideal point of the set $A^{t-1} \cup x$ and then calculates the Euclidean distance from each point in the archive to the ideal point. After that, it is removed the point with the highest distance. The hypothesis behind this method is that guiding the selection of the points in the archive to a region close to the ideal point will increase the convergence of the search to the Pareto front and will place the solutions in a good area of the objective space.
Distributed archiver (dist): This method also wants to direct the search to some regions in the Pareto front, but different from the Ideal archiver, these regions are not defined by only a single point. This method, tries to introduce convergence in the archive by guiding the archiving through reference points, but also tries to introduce diversity by selecting different area of the objective space. The basic idea, is to select the reference points (solutions closest to each dimension in the objective space and to the ideal point) and then to select the closest solutions to these points into proportional sets.

So, the \(f_{\text{filter}}(A^{t-1}, x)\) function first calculates the number of solutions selected for each region: total of solutions \((N + 1)\) divided by the number of dimension (number of objectives) plus 1 (the region closer to the ideal point). After that, for each dimension, every set is filled with the closest points to each reference point, until the set is full. At the end, the solution that is farthest from each reference point is discarded and \(A^t\) is defined as the union of each reference set.

Distance to Reference Points Archiver (eucli and tcheb):
The last method is similar to the Distributed archiver, but it does not fix the number of points selected for each reference point (the points closer to each dimension and the ideal point). The idea is to guide the archive to different regions from the objective space, but instead of fixing these regions, they are defined according to the search of the algorithm. So the \(f_{\text{filter}}(A^{t-1}, x)\) calculates the reference points, after, for every point in \(A^{t-1} \cup x\) the smaller distance for each reference point is calculated. Then, it is discarded the solution with the higher smaller distance. This method has the goal to identify a set of solutions that are closer to the reference points and then guide the archive to different regions of the objective space. For this method it was defined two different archivers: one that uses the Euclidian distance (eucli) to define the smaller distance and the other that uses the Tchebycheff (tcheb) distance.

Next section will perform an empirical analysis to observe the influence of each archiving method into a MOPSO algorithm for MaOPs. This study has the interest to observe how methods that focus on diversity or convergence behave in many-objective scenarios.

V. EXPERIMENTS
To evaluate the archiving methods it was used the SMPSO algorithm as basis [4], described in Section III. The 12 archivers discussed in the previous section were used. The size \(N\) of the archive was defined as 250 solutions. The algorithm executed 50000 fitness evaluation and the population was limited to 250 particles. In each iteration \(\omega\) varied randomly in the interval \([0, 0.8]\), \(\phi_1\) and \(\phi_2\) varied randomly in \([0, 1]\), \(C_1\) and \(C_2\) varied randomly over the interval \([1.5, 2.5]\). For the \(\epsilon\)-Pareto archiving methods, the value of \(\epsilon\) was defined as 0.0001, the same presented in [3].

The modified SMPSO was applied to one many-objective problems of the DTLZ family [8], namely DTLZ2. The DTLZ family is a set of benchmark problems often used in the analysis of MOEAs [4]. This problem can be scaled to any number of objectives \((m)\) and decision variables \((n)\) and the global Pareto front is known analytically. The DTLZ2 problem can be used to investigate the ability of the algorithms to scale up their performances in large numbers of objectives. In this study, we are interested in analyzing the behavior of the archivers in PSO for many objectives. So, in the empirical study, the archivers were applied to different problems and high dimensional objective spaces: 3, 4, 5, 6, 7, 8 and 9.

The experimental study investigates the behavior of the proposed approach, especially in terms of convergence and diversity, as well as the scalability with respect to the number of objectives. Considering this fact, the following quality indicators are used: Generational Distance (GD), Inverse Generational Distance (IGD) and Spacing [1]. Also it was used the Hypervolume indicator.

![Fig. 2: Mean of GD values for all 30 executions for all archivers](image)

The GD measures how far the approximation set \((PF_{\text{approx}})\) generated by each archiver is from the true Pareto front of the problem \(PF_{\text{true}}\). It is a minimization measure and it allows us to observe if the algorithm converges to some region in the true Pareto front. IGD measures the minimum distance of each point of \(PF_{\text{true}}\) to the points of \(PF_{\text{approx}}\). IGD allows us to observe if \(PF_{\text{approx}}\) converges to the true Pareto front and also if this set is well diversified. It is important to perform a joint analysis of GD and IGD indicators because if only GD is considered it is not possible to notice if the solutions are distributed over the entire Pareto front. On the other hand, if only IGD is considered it is possible to define a sub-optimal solution as a good solution. Spacing measures the range variance between neighboring solutions in the front. If the value of this metric is 0, all solutions are equally distributed in the objective space.

The Hypervolume is the most used quality indicator in Multi-Objective Optimization. However it has some limitations, including the computational cost for many-objectives and the privilege of sets that have concentrated solutions at the extremes of the Pareto Front [9]. Here, it was calculated
through the method described in [18]. However, the software was unable to calculate this indicator for the Unbound Archiver, for a number of objective greater than six objectives. For this reason, the analysis of the hypervolume was restricted only to 3, 4, 5, 6 number of objectives. Due to these limitations the analysis of the hypervolume will be considered here as a complement of the analysis of the other quality indicators.

Fig. 3: Mean of IGD values for all 30 executions for all archivers

One of the problems of the Many-Objective Optimization is the visualization of the approximation set. So, seeking to observe where the approximation set generated by each archiver is located, it is made the analysis of the distribution of Tchebycheff distance, proposed in [9]. It’s commonly accepted that decision makers often select a solution located in the middle of the Pareto front, called as "knee" of the Pareto front. The knee can be considered as the point with minimum distance to the ideal point [9]. The methodology used compares the Tchebycheff distance of each point of the \( P_{\text{approx}} \) to the knee. The distributions of the Tchebycheff distance for all solutions are presented in distribution charts, for all analyzed objectives.

Every algorithm was executed thirty times. The quality indicators are compared using the Friedman test [19] at 5% significance level. The test is applied to raw values of each metric. The post-test of the Friedman test indicates if there is any statistically difference between each analyzed data set, to identify which data set has the best values it is used some boxplot charts. The boxplot gives information about the location, spread, skewness and tails of the data.

VI. RESULTS AND DISCUSSIONS

For the GD, it can be observed in Table I that the \( \epsilon \)-Pareto approaches, the unbound archiver and the Ideal archiver have the best convergence towards the Pareto front. In Figure 2, it can be observed that the GD values of these methods do not deteriorate when the number of objective grows. In these methods, the values of GD for high number of objectives are similar to the low ones. The proposed method, Ideal Archiver, obtained the best average results. The results of the \( \epsilon \)-Pareto approaches are very similar to the Unbound archiver. As the \( \epsilon \)-Pareto approaches did not have a limited archive size, due to the low \( \epsilon \) value used they obtained similar results to the Unbound archiver. Besides, for a low number of objective almost all archivers have similar GD values, however, when the number of objective grows some methods suffer a high decay, specially the random archiver, the distributed archiver, the CD archiver and the AG archiver. It can be noted that worst results were obtained by methods that emphasize the diversity in the selection of the solutions that remain in the archiver.

For the IGD, again the Unbound archiver and \( \epsilon \)-Pareto approaches obtained good results. In addition to these methods, the dominating archiver and the MGA archiver also obtained the best results, according to the Friedman test. The good IGD values of these methods represent that these methods generate an approximation set distributed over the Pareto front. It can be highlighted the good results presented by the Unbound archiver and the \( \epsilon \)-Pareto approaches, that obtained a good convergence through the Pareto front and generated a distributed approximation set. By the dominating archive results, it can be observed that it is possible to cover a large area of the Pareto front only selecting solutions that dominate any other in the archive and the results presented by the MGA showed that the box method proposed by this archiver actually guide the algorithm to search a more diverse approximation set. However these two methods had a deterioration in the convergence for many objectives (deterioration of the GD values). So, they obtained a distributed approximation set but no so close to the Pareto front. In Figure 3, it can be observed that the worst IGD values were obtained by the CD archiver, the Random archiver and the Ideal archiver. The Ideal archiver obtained a low value of IGD, because this method does not distribute its solution, but instead concentrates the solutions near to the ideal point.

1 Available in http://iridia.ulb.ac.be/~manuel/hypervolume
TABLE I: Best archivers according to Friedman test for DTLZ2 problem

<table>
<thead>
<tr>
<th>Obj</th>
<th>GD</th>
<th>IGD</th>
<th>Spacing</th>
<th>Hypervolume</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and spea2</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>4</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>5</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>6</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>7</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>8</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
<tr>
<td>9</td>
<td>eapp, eaps, ideal and ub</td>
<td>dom, eapp, eaps, mga and ub</td>
<td>crowd, ideal and eapp</td>
<td>dom, eapp, eaps, mga and ub</td>
</tr>
</tbody>
</table>

For the Spacing, the best values were obtained by the CD, Ideal and MGA (Figure 4). Both CD and MGA focus on the diversity of the archive. These methods granted the diversification on the selected solutions and generated a good spacing between these solutions when the number of objective grows. The Ideal archive also obtained the best average spacing values, specially for high number of objectives. Since it limits its search to a small area of the objective space, near to the ideal point, it became easier for the PSO algorithm control the spacing of the solutions.

For the Hypervolume, the best archivers are the same methods that obtained the best IGD values: Dominating, Unbound, $\epsilon$-Pareto approaches and MGA archivers. In this comparison, the Ideal archiver, obtained Hypervolume values smaller than the other methods. This archiver obtained the best convergence but concentrates its solutions near the ideal point and then obtained the worst hypervolume. The best hypervolume were obtained by the archivers that focus on to diversify the approximation set through the Pareto front. If only we analyze the Hypervolume, methods that have good convergence properties are discarded, however, to obtain convergence towards the Pareto front is one of the main problems in Many-Objective Optimization.

Finally, the distributions of the Tchebycheff distance are analyzed. For clarity, only the method that had best results according the Friedman test in GD or IGD were plotted in the chart. In these charts, curves that have peaks near small values of Tchebycheff distance concentrated their solutions near the knee. For small number of objectives, almost all archivers have a similar distribution, often near the knee. However when the number of objective grows, some method tend to spread their distribution in different regions of the Pareto front, meanwhile others concentrate near the knee. As discussed in the analysis of the quality indicators, the Ideal archive has the goal to converge to a region near to the ideal point. This behavior can be observed through its distribution. In spite of the number of objectives, this method concentrates its solutions near the knee. The other methods, like the Unbound, the MGA, Dominating and the $\epsilon$-Pareto approaches presented similar distributions. When the number of objective grows, their distribution tend to have smaller peaks at small distance and to generate a more stretched distribution.

Fig. 5: Tchebycheff distance distribution for the best archivers

In summary, it is possible to control the convergence of a MOPSO algorithm applied to Many-Objective Problems by only using different archiving methods. The proposed method, Ideal Archiver, proved its hypothesis that is possible to obtained good convergence by guiding the selection of the points in the archive to a region close to the ideal point. Also, it can be observed that although the ideal point is used as a guide, the generated approximation set is not limited by only
one solution, since the PSO has other mechanism to introduce diversity on the search, like the leader selection. The other proposed method did not obtain good results. The strategy of spreading the search for the different dimensions can deliver distributed solutions but does not help the convergence to the Pareto Front. The approaches that do not limit the archive size, Unbound Archive and $\epsilon$-Pareto approaches, showed that the PSO can get good results with MaOPs, but it is not always possible to run the algorithm without an archive limit, since the number of solutions can grow too much. Finally the CD archiver obtained the worst result in terms of convergence and diversity showing that the SMPSO algorithm is not suitable for Many-Objective Optimization.

VII. CONCLUSION

Many-Objective Optimization Problems are still under explored in literature. Besides, some important features of PSO algorithms, like the different methods to store the leaders in the external archive, have not yet been explored in the literature. In the context of archiving methods, some works in literature already studied the influence of these methods in the search of Multi-Objective Evolutionary Algorithms. However, there was no work that studied how these archiving methods behave when the number of objective grows. This paper presented the analysis of different archiving methods applied to MOPSO in Many-Objective Optimization.

In this work, different archiving methods were selected and applied to a MOPSO algorithm in Many-Objective scenarios. The main goal was to observe how each method influences the MOPSO algorithm in terms of convergence and diversity over the Pareto front. Besides, some new methods, with specific characteristics of convergence or diversity, were proposed. To evaluate the archivers into MOPSO an empirical analysis was performed.

It can be concluded that the best results were obtained by archiving methods that do not limited the archive size. The Unbound Archiver and $\epsilon$-Pareto approaches obtained good convergence and diversity over the Pareto front. These good results of the $\epsilon$-Pareto approaches were first observed in [20] and has been discussed [6]. However, it is not always possible to run the algorithm without an archive limit. Also, the Ideal Archiver presented the best convergence towards the Pareto front. This method guided the MOPSO search to a region near to the knee of the Pareto front. However, the search was limited to this region and the Ideal archiver did not obtain a good distribution over the Pareto front, resulting in a low value of Hypervolume. The other proposed methods did not obtain good results and the archiving method used by the SMPSO algorithm obtained the worst results in MaOPs.

Future works include explore other characteristics of MOPSO algorithm, like the selection of the leaders. Besides, the MOPSO algorithm with different archiving methods must be compared to other algorithms specific proposed to MaOPs. Finally, the archivers will be analyzed in others benchmarking problems, such as discontinuous Pareto front.

REFERENCES