International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 3 (2011) pp.096-107

Using Different Many-Objective Techniques in Particle Swarm Optimization for Many Objective Problems: An Empirical Study

Andre B. de Carvalho and Aurora Pozo

Computer Science Department, Federal University of Parana,
Cel. Francisco H. dos Santos Street, Centro Politecnico, Curitiba, PR, Brazil
{*andrebc, aurora*}@*inf.ufpr.br*

Abstract:
Pareto based Multi-Objective Evolutionary Algorithms face several problems when dealing with a large number of objectives. In this situation, almost all solutions become non-dominated and there is no pressure towards the Pareto Front. The use of Particle Swarm Optimization algorithm (PSO) in multi-objective problems grew in recent years. The PSO has been found very efficient in solve Multi-Objective Problems (MOPs) and several Multi-Objective Particle Swarm Optimization algorithms (MOPSO) have been proposed. This work has the goal to study how PSO is affected when dealing with Many-Objective Problems. Recently, some many-objective techniques have been proposed to avoid the deterioration of the search ability of multi-objective algorithms. Here, two many-objective techniques are applied in PSO: Controlling the Dominance Area of Solutions and Average Ranking. An empirical analysis is performed to identify the influence of these techniques on convergence and diversity of the MOPSO search in different many-objective scenarios. The experimental results are analyzed applying some quality indicators and some statistical tests.
Keywords: Many-Objective Optimization, Multi-Objective Optmization, Particle Swarm Optmization

I. Introduction

Particle Swarm Optimization (PSO) [18] is a population based meta-heuristic that has been used to solve several optimization problems [24]. PSO algorithms are inspired on animal swarm intelligence and are based on the cooperation of the individuals. Multi-objective optimization problems (MOPs) are usually solved by a large number of multi-objective evolutionary algorithms (MOEAs) [19], including PSO. However, several problems arise when dealing with a large number of objectives. Problems with more than three objectives are called Many-Objective Problems (MaOPs) [14]. The main obstacle faced by MOEAs in many-objective is the deterioration of the search ability, because almost all solutions are non-dominated there is no pressure towards the Pareto Front.

To overcome these limitations, in recent years the interest for Many-Objective Optimization has grown [14] [25]. In this area, some techniques have been proposed like Controlling of Dominance Area of Solutions (CDAS) [26] and the use of rankings [17], like Average Ranking [3]. The goal of these works is to study techniques that decrease the negative effects of using several objectives.

The paper [9] presented a first study on the influence of the CDAS technique in PSO for Many-Objective Problems (MaOPs). The main idea of the paper was to observe the influence of controlling the dominance area of solutions on aspects like convergence and diversity in a metaheuristic based on cooperation between individuals. Here, this work presents an extension of the use of many-objective techniques in Particle Swarm Optimization. We extend the previous study by applying a different many-objective technique, called the Average Ranking (AR) [3]. This technique induces a preference ordering over a set of solutions, and then a MOPSO algorithm chose the best solutions through this preference order, instead of, dominance relation.

We perform an empirical analysis to measure the performance of MOPSO in Many-Objective problems using these two preference relations: CDAS and AR. The chosen algorithm is the SMPSO [22], and two extended algorithms are implemented CDAS-SMPSO [9] and AR-SMPSO. These algorithms are applied to two benchmark many-objective problems, DTLZ2 and DTLZ4 [11]. A set of quality indicators is used to investigate how these techniques affect convergence and diversity of MOPSO search in many objective scenarios : Generational Distance (GD), Inverse Generational Distance (IGD), Spacing and also it is analyzed the distribution of the Tchebycheff distance over the ”knee” of the Pareto front [15].

The rest of this paper is organized as follows: Section II presents the main concepts of many-objective optimization and Section III discusses some related works. In Section IV the previous work that uses CDAS in SMPSO is revised. After, Section V describes the use of AR in the SMPSO algorithm. Finally, Section VI presents empirical experiments and Section VII discusses the conclusions and future works.

II. Multi-Objective Optimization

Real world problems usually include multiple criteria that should be satisfied at the same time. Furthermore, in such problems, the objectives (or criteria) to be optimized are usually in conflict, i.e. trying to improve one of them will result in worse values for some other. For example, most decision maker is faced with a difficult decision problem; they want to assure a great level of reliability and also a minimum cost. In this case, the goal is to find a good "trade-off" of solutions that represent the better compromise among the objectives. The general multi-objective maximization problem (MOP) can be stated as in (1).

$$Maximize f(x) = (f_1(x), f_2(x)..., f_m(x)) \quad (1)$$

subject to $x \in \Omega$
where: $x \in \Omega$ is a feasible solution vector, Ω is the feasible region delimited by the constraints of the problem, and m is the number of objectives.
Important concepts used in determining a set of solutions for multiobjective optimization problems are dominance, Pareto optimality, Pareto set and Pareto front. Pareto Dominance (PD) was proposed by Vilfredo Pareto [23] and is defined as follows: given two solutions $x \in \Omega$ and $y \in \Omega$, for a maximization problem, the solution x dominates y if

$$\forall i \in \{1, 2, ..., m\} : f_i(x) \geq f_i(y), and$$
$$\exists i \in \{1, 2, ..., m\} : f_i(x) > f_i(y)$$

x is a non-dominated solution if there is no solution y that dominates x.
The goal is to discover solutions that are not dominated by any other in the objective space. A set of non-dominated solutions is called Pareto optimal and the set of all non-dominated objective vectors is called Pareto front. The Pareto optimal set is helpful for real problems, e.g., engineering problems, and provides valuable information about the underlying problem. In most applications, the search for the Pareto optimal is NP-hard and then the optimization problem focuses on finding an approximation set, as close as possible to the Pareto optimal. Multi-Objective Evolutionary Algorithms have been successfully applied to many MOPs. MOEAs are particularly suitable for this task because they evolve simultaneously a population of potential solutions to the problem obtaining a set of solutions to approximate the Pareto front in a single run of the algorithm.

III. Related Work

In a scalar objective optimization problem, all the solution can be compared based on their objective function values and the task of a scalar objective evolutionary algorithm is to find one single solution. However, in MOP, domination does not define a complete ordering among the solutions. Therefore, MOEAs modify Evolutionary Algorithms (EAs) [6] [13] 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. Although, most of the studies on MOPs have been 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 [14]. MOPs having more than 3 objectives are referred as many-objective optimization problems in the specialized literature. Several studies have proved that MOEAs scale poor in many-objective optimization problems. The main reason for this is that the proportion of non-dominated solutions in a population increases exponentially with the number of objectives. As consequence: The search ability is deteriorated because it is no possible to impose preferences for selection purposes; The number of solutions required for approximating the entire Pareto front also increases, and difficulty of the visualization of solutions.
Currently, the research community has been tackled these issues using mainly three approaches:

- Adaptation of preference relations that induce a finer order on the objective space [26], [1], [11], [17], [15], [25], [7],.
- The dimensionality reduction is also an alternative for dealing with the challenges of many objectives [21], [5], [16],[4]. The overall idea of this approach is to identify the least non conflicting objectives (one that can be removed without changing the Pareto optimal set) to discard them, for instance, dismissing objectives that are highly correlated with others.
- Decomposition strategies that uses decomposition methods, which have been studied in the mathematical programming community, into evolutionary algorithms for multi-objective optimization. This approach decomposes the MOP into a number of scalar optimization problems, and then, evolutionary algorithms are applied to optimize these sub problems simultaneously [28], [2].

In sum, these works reflect the focus of the current research when dealing with many-objective optimization problems (MaOPs). One of the main conclusions of these works is related to the weakness of the Pareto dominance relation for dealing with MaOPs and some alternative were proposed. Some authors point out that by using an effective ranking scheme, it is possible for MOEAs to converge in MaOPs. But, the ranking method must provide a fine grained discrimination between solutions. On the other hand, a high selection pressure sacrifices diversity and the algorithm converges to a small region.
So, there exist many difficulties waiting to overcome and motivate our work. One of them is related to the metaheuristic, until relatively recently, most of the research had concentrated on a small group of algorithms, often the NSGA-II. In this work, the behavior of the Particle Swarm Optimization in MaOPs is investigated. Two previous works deal with MaOPs using PSO algorithms.
In [27], it is presented an approach that uses a distance metric based on user-preferences to efficiently find solutions. In the work, the user defines good regions on the objective space that must be explored by the algorithm. So, PSO is used as a baseline, and the particles update their position and velocity according to their closeness to the preference regions. In this method, the PSO algorithm does not rely on Pareto dominance comparisons to find solutions. The algorithm was compared to a user-preference based PSO algorithm that uses Pareto dominance comparisons to select the leaders. The results showed that the algorithm obtain better results, especially for problems with high number of objectives. In [20] a PSO algorithm handles with many-objectives using a Gradual Pareto dominance relation to overcome the problem of

finding non-dominated solutions when the number of objectives grows.

As explained before, one of the alternative to deal with MaOPs is to use an effective ranking scheme, however, these ranking schemes have never been object of study with PSO. Then, differently of the previous many-objective PSO works, our work has the goal to study the behavior of preference relations in Multi-Objective Particle Swarm algorithm, a topic few explored in the literature. The selected technique was the Control of Dominance Area of Solutions (CDAS) [26], and it was first applied to PSO in [9]. Here we extend this previous work and we apply other many-objective technique to PSO, the Average Ranking. The two techniques are also compared in different many-objective situations.

IV. Previous Work

This Section reviews our previous work [9] that had the goal to apply the control of dominance area technique (CDAS) into PSO algorithm. First, the CDAS technique is described and its application into PSO algorithm. Finally, some results are discussed.

Sato *et al.* propose a method to control the dominance area of solutions to induce an appropriate ranking of the solutions. The proposed method controls the degree of contraction and expansion of the dominance area of solutions using a user-defined parameter S_i. The dominance relation changes with this contraction or expansion, and solutions that were originally non-dominated become dominated by others. The modification of the dominance area is defined by the Equation (2):

$$f_i'(x) = \frac{r \cdot sin(\omega_i - S_i\pi)}{sin(S_i\pi)} \quad (2)$$

where x is a solution in the search space, $f(x)$ is the objective vector and r is the norm of $f(x)$. ω_i is the degree between $f_i(x)$ and $f(x)$. If $S_i = 0.5$ then $f_i'(x) = f_i(x)$ and there is no modification in the dominance relation. If $S_i < 0.5$ then $f_i'(x) > f_i(x)$, so will be produced a subset of the Pareto Front. In the other hand, if $S_i > 0.5$ then $f_i'(x) < f_i(x)$ and the dominance relation is relaxed, so solutions that were normally dominated become non-dominated.

PSO is a population-based heuristic inspired by the social behavior of bird flocking aiming to find food [18]. In PSO, the system initializes with a set of solutions and search for optima by updating generations. The set of possible solutions is a set of particles, called swarm, which moves in the search space, in a cooperative search procedure. These moves are performed by an operator that has a local and a social component. This operator is called velocity of a particle and moves it through an n-dimensional space based on the best positions of their neighbors (social component), the leader, and on their own best position (local component). The best particles are found based on the fitness function. There are many fitness functions in Multi-objective Particle Swarm Optimization (MOPSO). Based on Pareto dominance concepts, each particle of 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 repository (archive) [18], which contains the best non-dominated solutions found so far.

The chosen MOPSO algorithm was the SMPSO. The SMPSO algorithm was presented in [22]. In this algorithm,

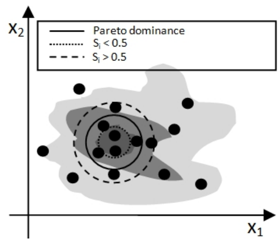


Figure. 1: Influence of the CDAS in MOPSO leader's choice.

the velocity of the particle is limited by a constriction factor χ. The SMPSO introduces a mechanism that bound the accumulated velocity of each variable j (in each particle). Besides, after the velocity update of each particle a mutation operation is applied. It is applied a polynomial mutation in 15% of the population, randomly selected. In the SMPSO, the archive of the leaders has a maximum size, defined by a user parameter. When this archive becomes full, the crowded distance is used [22] to define which particles will remain in the repository. The choice of the leader is defined by a binary tournament.

In MOPSO convergence and diversity are controlled by the cooperation between the particles, i.e., the choice of the leaders. The leaders guide the swarm to the best areas in the search space, so depending on the choice of the leaders the solutions can converge for a small area of the Pareto Front or perform a diversified search, trying to cover a larger region of the Pareto front. In MOPSO literature there are several methods to perform this selection, e.g., the sigma distance, a simple binary tournament, among others [24].

In the previous work, we studied the influence of the CDAS technique in a MOPSO algorithm for many-objective scenarios. This technique was incorporated in the search as follows: as the Sato *et al.* technique modifies the dominance relation, the step that updates the non-dominated archive was modified and now applies the new dominance relation defined by Equation (2). Figure 1 presents an example of the application of CDAS in a 2-dimensional search space. The darker areas represent the best areas in the search space, where all solution should converge. In the MOPSO algorithm the leaders will be the particles near to these areas. The selected leaders by the original Pareto dominance relation are represented by the solid circle. When the dominance area is modified by the CDAS with a $S_i < 0.5$ (dotted circle), less solutions become non-dominated and the algorithm tends to converge to a small area of the search space and to decrease the diversity. For $S_i > 0.5$ (dashed line), new solutions that were dominated become non-dominated and now influence the others particles in the swarm. In this situation the algorithm tends to diversify its search, but as the original non-dominated solutions still influence the particles of the swarm, the algorithm still has the characteristic to converge to the best area of the search space. This algorithm was called CDAS-SMPSO.

A. Previous experiments

The CDAS-SMPSO was used in the DTLZ2 problem of the DTLZ family [11]. It was performed an empirical analysis that applied CDAS-SMPSO in different and large objective values: 3, 5, 10, 15 and 20 objectives. The parameter that controls the degree of S_i was defined for eleven different values. The S_i value varied in intervals of 0.05, the same variation applied in [26]. S_i varies in the range $[0.25, 0.45]$, for the selection of a subset of the Pareto Front, varies in the range $[0.55, 0.75]$ for the relaxation of the dominance relation and has the value equal to 0.5 for the original Pareto dominance relation.

Some quality measures were used to observe how convergence and diversity are affected by the control of the dominance area of the solutions in PSO [29]. The Generational Distance (GD) was used to observe if the algorithm converges for some region of the true Pareto Front. The Inverse Generational Distance to observe if the PF_{approx}, i.e, the solutions generated by the CDAS-SMPSO algorithm, converges to the true Pareto front and also if this set is well diversified. The variance of the distance between neighbors solution in the front is measured by the Spacing. For the comparison of the quality indicators the Friedman test at 5% significance level is used. The Friedman test is a non-parametric statistical test used to detect differences between algorithms [12]. The test is applied to raw values of each metric.

Besides, it was performed a comparison discussed in [15] to observe the convergence and diversity for each configuration. In literature, it is accepted that decision makers prefer solutions in the middle of the Pareto front, called the "knee" of the Pareto front. So, in [15] it is presented a methodology that compares the distance of each point of the PF_{approx} to the knee, the Tchebycheff distance.

Table 1 presents the summary of the Friedman test. For each objective value, the configuration with best results for each quality measure is presented.

GD best results were obtained when $S_i < 0.5$. The configurations $S_i = 0.25$, 0.3 0.35 and 0.4 obtained the best results for almost all objectives, i.e., obtained a statistically significant difference with respect to other configurations. Besides, it was observed that when the number of objectives grows the convergence of the original dominance relation deteriorates, achieving poor GD results. For the IGD, according to the Friedman test, the results of the configurations $S_i = 0.3$, 0.35, 0.4 and 0.45 had the best results. Again, for IGD when the number of objectives is small, the SMPSO with the original Pareto dominance relation still has competitive IGD results; however, when this number grows its performance deteriorates. It was concluded examining these two indicators that the CDAS with $S_i < 0.5$ produced very good results for many objectives.

For the spacing indicator, the best configuration according to the Friedman test was the extreme $S_i = 0.25$. However, this occurs because for almost all objective this configuration generated only one solution in the PF_{approx}. Again, the configuration with the original Pareto dominance relation obtained the worst results.

For the Tchebycheff analysis, the original dominance produced distributions that were not concentrated in any region, generating equivalent distributions for different distances. These distributions reflect the results of GD and IGD. The SMPSO with the original Pareto dominance relation do not converge to the true Pareto Front, but, it generates a distributed PF_{approx} because its IGD values are low, consequently, it generates a sub-optimal solution. The best distributions were the configurations $S_i = 0.35$, 0.4 and 0.45. These configurations concentrated almost all its solutions in a small distance of the knee, even when dealing with high number of objectives. These concentrated distributions stress the GD and IGD results and show that the CDAS with $S_i < 0.5$ improves the convergence of the PSO algorithm. For low values of S_i, only few solutions remain non-dominated and the algorithm converges for a small region, often close to the knee. For configurations with $S_i > 0.5$, it was concluded that high values of S_i produces a diversified PF_{approx}. With a degree near to the original Pareto dominance relation, $(0.5 < S_i \leq 0.65)$, the solutions were concentrated in a region with small values of the Tchebycheff distance. However, the CDAS with $S_i > 0.5$ did not have the same power of convergence than $S_i < 0.5$.

V. Ranking Based PSO

In the previous work [9], the influence of the CDAS technique into PSO algorithm was analyzed using the SMPSO [22]. This paper extends this work by analyzing the behavior of another many-objective technique into SMPSO. The chosen method is the Average Ranking (AR) that according to [7] produced the best results among different ranking methods. In this section, first, the main aspects of the PSO are discussed and details about SMPSO are given. Finally, the AR and its implementation into SMPSO algorithm are described. Here, the implementation of AR into SMPSO is called AR-SMPSO.

A. SMPSO

Particle Swarm Optimization is a population-based heuristic inspired by bird. PSO performs a cooperative search procedure between the solutions. The set of possible solutions is a set of particles, called swarm, which moves in the search space through the velocity operator that is based on the best positions of their neighbors (social component), the leader, and on their own best position (local component).

Multi-objective particle swarm optimization uses Pareto dominance concepts to define the leaders. Each particle of the swarm could have different leaders, but only one may be selected to update the velocity. The basic steps of a MOPSO algorithm are: initialization of the particles, computation of the velocity, position update, mutation and update of leader's archive.

Each particle p_i, at a time step t, has a position $x(t) \in R^n$ (3), that represents a possible solution. The position of the particle, at time $t + 1$, is obtained by adding its velocity, $v(t) \in R^n$ (4), to $x(t)$:

$$\overrightarrow{x}(t+1) = \overrightarrow{x}(t) + \overrightarrow{v}(t+1) \quad (3)$$

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. The velocity is defined as follows:

$$\overrightarrow{v}(t+1) = \varpi \cdot \overrightarrow{v}(t) + (C_1 \cdot \phi_1) \cdot (\overrightarrow{p}_{best}(t) - \overrightarrow{x}(t)) + (C_2 \cdot \phi_2) \cdot (\overrightarrow{R_h}(t) - \overrightarrow{x}(t)) \quad (4)$$

Table 1: Best configurations for CDAS-SMPSO algorithm, DTLZ2 problem.

Problem	Objective	GD	IGD	Spacing
DTLZ2	**3**	0.35, 0.4 and 0.45	0.45, 0.5 and 0.55	0.6 and 0.65
	5	0.3, 0.35 and 0.4	0.35, 0.4 and 0.45	0.25, 0.6 and 0.65
	10	0.25, 0.3 and 0.35	0.3, 0.35 and 0.4	0.25, 0.6 and 0.65
	15	0.25, 0.3 and 0.35	0.3, 0.35 and 0.4	0.25 and 0.3
	20	0.25, 0.3 and 0.35	0.25, 0.3, 0.35 and 0.4	0.25, 0.3 and 0.35

The variables ϕ_1 and ϕ_2, in (4), are coefficients that determine the influence of the particle best position, $\overrightarrow{p_{best}}(t)$, and the particle global best position, $\overrightarrow{R_h}(t)$. The constants C_1 and C_2 indicates how much each component influences on velocity. The coefficient ϖ 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. The repository of leaders is filled with the best particles after all particles of the swarm were updated.

The SMPSO algorithm was presented in [22]. This algorithm has the characteristic to limit the velocity of the particles. In this algorithm the velocity of the particle is limited by a constriction factor χ, that varies based on the values of C_1 and C_2. Besides, the SMPSO introduces a mechanism that bound the accumulated velocity of each variable j (in each particle) by applying the Equations (5), (6), (7)and (8) (the upper and lower limits are parameters defined by the user).

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}} \tag{5}$$

$$\varphi = \begin{cases} C_1 + C_2 & \text{if } C_1 + C_2 > 4, \\ 1 & \text{if } C_1 + C_2 \leq 4. \end{cases} \tag{6}$$

$$\varphi = \begin{cases} delta_j & \text{if } v_{i,j}(t) > delta_j, \\ -delta_j & \text{if } v_{i,j}(t) \leq -delta_j, \\ v_{i,j} & \text{otherwise.} \end{cases} \tag{7}$$

$$\chi = \frac{upper\_limit_j - lower\_limit_j}{2} \tag{8}$$

After the velocity update of each particle a mutation operation is applied. It is applied a polynomial mutation [10] in 15% of the population, randomly selected. The leader is chosen by a binary tournament. In the SMPSO, the leader's archive has a maximum size, defined by a user parameter. The crowded distance [10] defines which particles will remain in the repository when the archive becomes full.

B. Average Ranking

The Average Ranking method was proposed in [3]. This technique is a preference relation that induces a preference ordering over a set of solutions. The AR independently computes a ranking for each objective value. After the computation of each ranking, the AR is the sum all these rankings. The AR can be simple defined for a solution S by Equation (9):

$$AR(S) = \sum_{1<i<m} ranking(f_i(S)) \tag{9}$$

where m is the number of objectives and $ranking(f_i(S)$ is

Table 2: Average Ranking example

(f_1,f_2,f_3)	f_1	f_2	f_3	**AR**
(9, 1, 3)	4	1	2	7
(4, 2, 6)	3	2	4	9
(1, 7, 7)	1	4	5	10
(2, 8, 1)	2	5	1	8
(7, 5, 8)	5	3	6	14
(9, 9, 4)	6	6	3	15

```
Initialization procedure.
Evaluation of all objectives for all particles.
Initialization of repository and leaders choice.
Evolutionary loop
    Velocity and position updated for all particles.
    Mutation of 15% particles.
    Objectives evaluation for all particles.
    Update of the repository by using AR method.
    Calculation of crowding distance.
    Repository's prune (Best AR and CD).
    Selection of the non-dominated solutions in the repository.
Return the particles in the repository.
```

Figure. 2: AR-SMPSO algorithm.

the ranking for the ith objective. Table 2 presents one example of the AR. First, each solution is ranked by each objective, e.g., the third solution has the best ranking (ranking 1) for the first objective. After the calculation of each ranking the AR sum all the rankings and defines its value for each solution.

As defined in [15], here it will be used a different preference relation instead of the Pareto dominance relation: a solution x dominates solution y with respect to the average relation, denoted $x \prec_{AR} y$, if and only if $AR(x) < AR(y)$.

C. AR-SMPSO

AR-SMPSO algorithm uses the dominance relation $\prec_{AR}$ to enhance the SMPSO algorithm to deal with many-objective problems. Figure 2 presents a brief description of AR-SMPSO. First, an initialization procedure is performed. This procedure initiates all components of a particle (position, velocity, local leader, etc.). After, all objectives are evaluated. Then, the external archive (repository) containing the best solutions must be initiated. Here, to perform an initial pressure towards the best solutions only 10% of the solutions fills the repository. These solutions have the best AR values. After this step, the leaders are chosen though a binary tournament. The next step of AR-SMPSO is the evolutionary loop. In this loop, first the particles are moved through the search space by the velocity and position update and then it is applied a mutation in 15% of the particles. The update of the repository is the main difference from AR-SMPSO to the original SMPSO. In this step, the the $\prec_{AR}$ relation is applied. One solution enters in the repository if and only if it dominates any other solution with respect to the average relation. Then, the Crowded Distance (CD) is calculated for all solutions in the repository. As presented in Section V-A, the SMPSO algorithm limits the number of solutions in the repository. So, it must occur a prune procedure. This prune procedure keeps the solutions with best AR. If the AR is equal, then CD is used. Finally, to perform a pressure towards the Pareto front, only the nondominated solutions are kept in the external archive.

Next Section presents the empirical experiments performed to evaluate the AR-SMPSO algorithm. It is analyzed how the AR influences convergence and diversity of the SMPSO's search for many objectives problems. Besides, its results are compared to SMPSO and CDAS-SMPSO algorithms.

VI. Experiments

In this section, it is presented an empirical analysis to investigate the performance of PSO with many-objective techniques for many objective problems. Here, the two many-objective techniques discussed before are used: control of dominance area of solutions, called CDAS-SMPSO, and the Average Ranking, called AR-SMPSO.

The modified algorithms were applied to 2 many-objective problems of the DTLZ family [11], DTLZ2 and DTLZ4. The DTLZ family are a set of benchmark problems often used in the analysis of MOEAs [22] [27]. These problems were selected for this study because they share the following important features: a) the relatively small implementation effort(Bottom-up approach and constraint surface approach), b) can be scaled to any number of objectives (M) and decision variables (n), c) the global Pareto front is known analytically, d) convergence and diversity difficulties can be easily controlled. For each problem, the variable k represents the complexity of the search, where k = n - M + 1 (n number of variables, M number of objectives). The problems are built with non-overlapping sets of decision variables and $|\mathbf{x}_M|$ = k. The DTLZ2 problem can be used to investigate the ability of the algorithms to scale up its performance in large number of objectives. The DTLZ4 problem is used in order to investigate the ability to maintain a good distribution of solutions. In this study, we are interested to analyze the behavior of the modified PSO algorithms with the many-objectives techniques for many objectives. So, in the empirical study, the algorithms are applied to different problems and high dimensional objective spaces: 3, 5, 10, 15 and 20.

The same way as presented in the previous work [9], here, this experimental study had the goal to investigate the behavior of the proposed approaches, especially in terms of convergence and diversity, as well as their scalability with respect to the number of objectives functions. So, the this set of quality measures is used:

Generational Distance (GD) measures how far the generated approximated Pareto front PF_{approx}, i.e, the solutions generated by the algorithms, are from the true Pareto front of the problem PF_{true}. If GD is equal to 0 all points of PF_{approx} belong to the true Pareto front. GD allows observing if the algorithm converges for some region of the true Pareto Front.

Inverse Generational Distance (IGD) measures the minimum distance of each point of the PF_{true} to the points of the PF_{approx}. If IGD is equal to zero, the PF_{approx} contains every point of the true Pareto Front. IGD allows to observe if the PF_{approx} converges to the true Pareto front and also if this set is well diversified. It is important to perform a joint analysis of the GD and IGD indicators because if only GD is considered it is not possible to identify 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 [29] measures the range variance between neighbors solution in the front. If the value of this metric is 0, all solutions are equally distributed in the objective space.

We also compared the **Execution Time** of each algorithm and performed an analysis of the distribution of the Tchebycheff distance:

The distribution of the Tchebycheff distance is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension, Equation (10). Here, this distance is used to measures the minimum Tchebycheff distance of each approximation obtained to the ideal point, z., or the "knee" of the Pareto front. The distributions of the Tchebycheff distance for all solutions are presented in distribution graphs, for all analyzed objectives. The main motivation to use this metric is because in literature, it is accepted that decision makers prefer solutions in the middle of the Pareto front, called the "knee" of the Pareto front [8].

$$d(z, z^*, \lambda) = max_{1 \leq j \leq m}\{\lambda_j |z_j^* - z_j|\} \qquad (10)$$

z^* is the knee of the Pareto front, z is an objective vector in PF_{approx}, m is the number of objectives and $\lambda_j = 1/R_i$, where R_i is the range of the $j - th$ objective in the true Pareto Front.

For the CDAS-SMPSO, the parameter that controls de degree of S_i is defined for eleven different values, performing different configurations. The S_i value varied in intervals of 0.05, the same variation applied in [26]. S_i varies in the range $[0.25, 0.45]$, for the selection of a subset of the Pareto Front, varies in the range $[0.55, 0.75]$ for the relaxation of the dominance relation and has the value equal to 0.5 for the original Pareto dominance relation. The AR-SMPSO, did not need any specific parameter configuration.

Both algorithms were executed fifty times. All configurations, each S_i value and AR, were executed with 100 generations and 250 particles. ω varies in the interval $[0, 0.8]$, and both ϕ_1 and ϕ_2 vary in the range $[0, 1]$. C_1 and C_2 vary in the interval $[1.5, 2.5]$. The size of the repository is defined as the same size of the population. It is applied a polynomial mutation with probability $p_m = 1/n$, where n is the number of variables of the problem. Each variable of the velocity is limited to the range $[-5, +5]$. All parameters were defined with the values presented in [22].

Next Section describes the results of this empirical study. First, similar experiments to [9] are presented, only using the CDAS-SMPSO, but now using the DTLZ4 problem. These experiments compare each configuration of CDAS and analyze all quality indicators and then, the best configurations for each indicator are selected.

After, the results of AR-SMPSO were compared to each best configuration of CDAS-SMPSO and to the original dominance relation, $S_i = 0.5$, for all quality indicator. Also, the distribution of the Tchebycheff is discussed.

A. Results

Table 3 presents the best configurations obtained for the DTLZ4 problem, for each indicator. For GD, the configurations $S_i < 0.5$ exhibited best results. In this problem, only $S_i < 0.5$ had the best results, especially $S_i = 0.35$, 0.4 and 0.45. For IGD, the CDAS-SMPSO obtained similar results to DTLZ2 problem. Good results for $S_i < 0.5$, especially $S_i = 0.25$, 0.45 and 0.55. These results stress that using low values of S_i we can have good convergence and diversity in most cases, but there are some situations that the opposite can be true. For Spacing, it can be highlighted that good spacing results were obtained with low values of S_i. This often occurs due to the small number of solutions. The best spacing results were obtained with the configurations that generated smaller set of solutions.

Tables 4, 5, 6 and 7 present the results of the comparison between CDAS-SMPSO and AR-SMPSO. Each table presents the mean value of the indicators, for all executions. The cells marked with $*$ represent configurations that did not participate in the comparison, only best configurations of CDAS-SMPSO were selected.

For GD, Table 4, the CDAS-SMPSO outperformed the AR-SMPSO for all number of objectives, for both problems. It can be observed that when the number of objective grows the difference between the techniques decreases. Besides, the AR-SMPSO obtained best results than the original dominance relation for all comparisons. In sum, the CDAS technique obtained the best convergence, however AR-SMPSO also obtained good convergence and did not deteriorate when the number of objective grows.

Table 5 presents the IGD results. Again, the best results were obtained by the CDAS technique. For the DTLZ2 problem, the CDAS-SMPSO obtained the best results for all number of objectives, however the AR-SMPSO obtained very close results. The AR technique obtained better IGD than using the original dominance relation and its results did no deteriorate for high number of objectives. For DTLZ4, the CDAS-SMPSO obtained much better IGD than the others. As for the GD, the difference between AR-SMPSO and CDAS-SMPSO decreases when the number of objective grows. Again, AR-SMPSO outperformed the original dominance relation. Therefore, it can be concluded that CDAS-SMPSO generated a more distributed PF_{approx} than AR-SMPSO and both algorithms performed better than the original dominance relation considering convergence and distribution.

The results were similar for the Spacing indicator, presented at Table 6. CDAS-SMPSO obtained the best results, for both problems. This occurs due to the small number of solutions generated by this algorithm. As discussed in [9], the smaller the size of PF_{approx} is, the smaller is the spacing. The AR-SMPSO generated a constant number of solutions, often as big as the size of the repository. Again, the AR-SMPSO obtained better results than the original dominance relation.

Table 7 presents the average execution time, in seconds, for all CDAS-SMPSO configurations and AR-SMPSO. For DTLZ2, the CDAS-SMPSO obtained the best execution time, but only for the configuration with $S_i = 0.25$. This configuration obtained a low execution time due to the small PF_{approx}. The AR-SMPSO executed much faster than the others configurations, including the original dominance relation. For the DTLZ4, the results were different. The CDAS-SMPSO executed faster than the AR-SMPSO, especially when $S_i < 0.5$. This occurred because the CDAS-SMPSO generated a smaller PF_{approx}, for all objective studied. It is important to highlight, that this small set of solutions did not deteriorate the quality of the algorithms.

In summary, the CDAS technique was the best many-objective technique. It outperformed the AR for all indicators analyzed. The CDAS-SMPSO obtained better convergence and diversity than AR-SMPSO. However, AR-SMPSO obtained better results than the original dominance relation, especially when the number of objective grows. Also, AR-SMPSO does not require an extra parameter and this is an important advantage. It also executes faster, independently of the problem. For the CDAS-SMPSO, the user parameter S_i influences the quality of the search. This algorithm obtained best results with different values of S_i, however, it can be stated that S_i lower than 0.5 generates the best results. It was also showed that, the CDAS-SMPSO can execute faster, however this execution time is directly related to the number of solutions and there is no guaranteed that this algorithm will be faster for every problem.

B. Tchebycheff distribution analysis

Figures 3 and 4 present the distribution of the Tchebycheff distance for best configurations of CDAS-SMPSO, AR-SMPSO and SMPSO($S_i = 0.5$), for all number of objectives. Here, algorithms that have more solutions around the knee are better, i.e, the algorithm that concentrates its distribution in a smaller Tchebycheff distance.

For the DTLZ2 problem, the CDAS-SMPSO produced the best distribution. For 3 and 5 objectives, all configurations generated its distributions concentrated in low Tchebycheff values, i.e., near the knee. When the number of objectives grows, only the configurations $S_i = 0.25$ and 0.3 still concentrated its distributions near the knee. The other ones, produced a more diversified distribution. As presented in [9], the original dominance relation, $S_i = 0.5$, produced a similar distribution for almost all objectives, for both problems. The solutions of these configurations are distributed through different distances. Furthermore, these distributions were far from the knee. This same diversification of the distributions occurred for the AR-SMPSO.

For DTLZ4, similar results can be observed. Again, the original dominance relation and AR-SMPSO generated diversified distributions. However, AR-SMPSO concentrated its solutions farther from the knee than the original dominance relation. Again the best results were obtained by CDAS-SMPSO. For this problem, almost all generated solutions were concentrated at the same Tchebycheff distance, closer to the knee.

In sum, the CDAS-SMPSO was the best technique, now considering the distance from the knee. It generated almost all

Table 3: Best configurations for CDAS-SMPSO algorithm, for DTLZ4 problem.

Problem	Objective	GD	IGD	Spacing
DTLZ4	3	0.25, 0.3 0.35, 0.4 and 0.45	0.25, 0.3 and 0.7	0.25 and 0.75
	5	0.35, 0.4 and 0.45	0.25, 0.3 and 0.35	0.25 and 0.75
	10	0.35, 0.4 and 0.45	0.25, 0.3 and 0.35	0.25, 0.3 and 0.35
	15	0.35, 0.4 and 0.45	0.25, 0.3 and 0.35	0.25, 0.3 and 0.35
	20	0.35, 0.4 and 0.45	0.25, 0.3, 0.35 and 0.4	0.25, 0.3 and 0.35

Table 4: GD values for best CDAS-SMPSO configurations, original dominance and AR-SMPSO, for each number of objectives and for both DTLZ problems.

Prob	Obj	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	AR
DTLZ2	3	*	*	3.17E-03	2.45E-03	3.64E-03	1.59E-02	*	*	*	*	*	1.52E-02
	5	*	1.23E-02	9.85E-03	1.37E-02	*	6.42E-02	*	*	*	*	*	4.47E-02
	10	1.94E-02	2.48E-02	3.73E-02	*	*	1.32E-01	*	*	*	*	*	6.22E-02
	15	2.24E-02	2.67E-02	5.04E-02	*	*	1.60E-01	*	*	*	*	*	8.68E-02
	20	1.85E-02	2.82E-02	5.32E-02	*	*	1.75E-01	*	*	*	*	*	1.11E-01
DTLZ4	3	5.25E-05	6.01E-05	5.81E-05	6.48E-05	8.02E-05	9.60E-03	*	*	*	*	*	1,52E-02
	5	*	1.23E-02	9.85E-03	1.37E-02	*	6.42E-02	*	*	*	*	*	3.97E-02
	10	1.94E-02	2.48E-02	3.73E-02	*	*	1.32E-01	*	*	*	*	*	6.22E-02
	15	2.24E-02	2.67E-02	5.04E-02	*	*	1.60E-01	*	*	*	*	*	8.68E-02
	20	1.85E-02	2.82E-02	5.32E-02	*	*	1.75E-01	*	*	*	*	*	1.09E-01

Table 5: IGD values for best CDAS-SMPSO configurations, original dominance and AR-SMPSO, for each number of objectives and for both DTLZ problems.

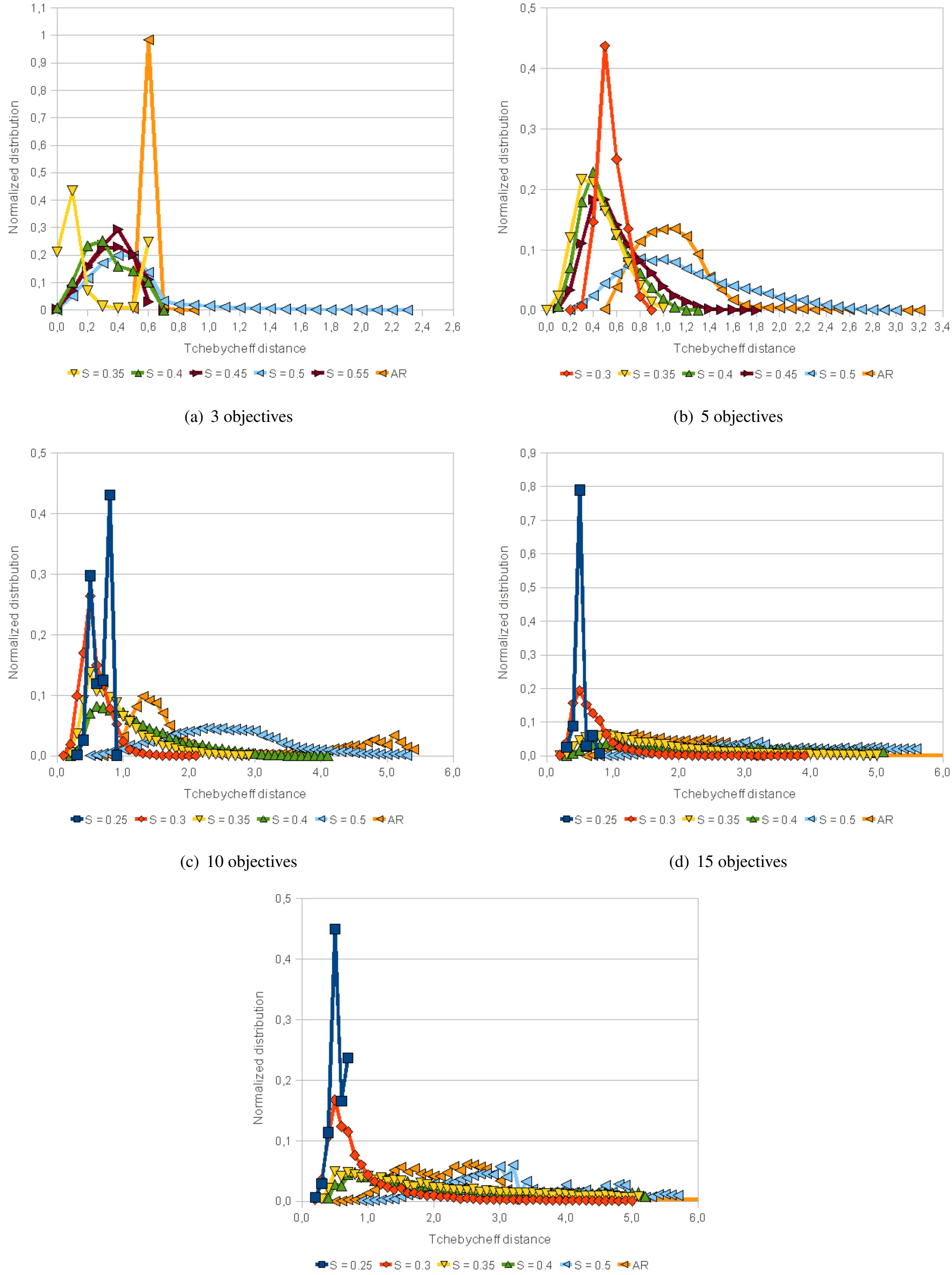
Prob	Obj	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	AR
DTLZ2	3	*	*	*	*	7.77E-04	9.46E-04	8.78E-04	*	*	*	*	1.84E-03
	5	*	*	2.85E-03	2.32E-03	2.47E-03	4.93E-03	*	*	*	*	*	4.56E-03
	10	*	3.61E-03	3.24E-03	3.69E-03	*	1.10E-02	*	*	*	*	*	5.54E-03
	15	*	3.24E-03	3.43E-03	4.12E-03	*	1.45E-02	*	*	*	*	*	5.77E-03
	20	4.84E-03	3.61E-03	3.24E-03	3.69E-03	*	1.10E-02	*	*	*	*	*	6.20E-03
DTLZ4	3	5.31E-03	7.06E-03	*	*	*	2.91E-02	*	*	*	1.15E-02	*	2.74E-03
	5	3.10E-04	3.87E-04	4.82E-04	*	*	1.37E+00	*	*	*	*	*	1.46E-03
	10	1.63E-06	2.36E-06	3.11E-06	*	*	2.04E+00	*	*	*	*	*	1.20E-03
	15	4.44E-08	8.21E-08	9.27E-08	*	*	2.18E+00	*	*	*	*	*	1.68E-03
	20	4.04E-09	7.56E-09	9.94E-09	1.44E-08	*	2.23E+00	*	*	*	*	*	1.87E-04

Table 6: Spacing values for best CDAS-SMPSO configurations, original dominance and AR-SMPSO, for each number of objectives and for both DTLZ problems.

Prob	Obj	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	AR
DTLZ2	3	*	*	*	*	*	8.08E-01	*	4.11E-01	2.11E-01	*	*	8.79E-01
	5	4.46E-01	*	*	*	*	1.53E+00	*	6.82E-01	1.51E-01	*	*	1.27E+00
	10	*	4.96E-01	2.39E+00	1.27E+00	*	8.78E-01	*	*	*	*	*	1.36E+00
	15	5.53E-01	7.57E-01	*	*	*	2.54E+00	*	*	*	*	*	1.82E+00
	20	5.52E-01	7.47E-01	9.84E-01	*	*	2.56E+00	*	*	*	*	*	2.28E+00
DTLZ4	3	5.31E-03	*	*	*	*	5.10E-01	*	*	*	*	8.76E-02	3.55E-01
	5	3.10E-04	*	*	*	*	1.37E+00	*	*	*	*	1.35E-01	5.73E-01
	10	1.63E-06	2.36E-06	3.11E-06	*	*	2.04E+00	*	*	*	*	*	8.50E-01
	15	4.44E-08	8.21E-08	9.27E-08	*	*	2.18E+00	*	*	*	*	*	8.31E-01
	20	4.04E-09	7.56E-09	9.94E-09	*	*	2.23E+00	*	*	*	*	*	7.73E-01

Table 7: Execution time (seconds) for all configurations of CDAS-SMPSO and AR-SMPSO, for each number of objectives and for both DTLZ problems.

Prob	Obj	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	AR
DTLZ2	3	0.52	1.44	6.42	34.32	39.05	32.88	43.44	43.60	42.22	35.82	61.73	4.72
	5	1.60	33.65	66.56	475.11	160.80	68.85	589.97	184.03	23.92	97.48	614.71	8.50
	10	7.37	758.33	144.74	1091.48	962.58	157.86	609.50	451.70	210.78	148.25	653.08	15.72
	15	12.61	626.27	446.05	1037.35	344.03	434.92	791.24	478.90	628.44	1010.35	776.83	23.48
	20	21.05	1078.43	602.08	979.36	865.95	1020.49	820.51	1182.44	815.63	731.59	954.37	32.77
DTLZ4	3	0.92	2.49	3.01	3.90	4.97	9.58	5.93	5.35	4.25	2.09	8.67	3.32
	5	2.16	5.20	6.25	7.85	10.10	17.70	9.81	7.96	5.55	4.21	11.01	8.76
	10	3.56	5.91	7.36	8.00	11.53	35.22	11.98	10.09	8.23	9.60	16.99	19.95
	15	10.73	16.18	16.89	21.09	25.25	124.62	34.04	28.52	24.65	25.27	60.65	30.22
	20	19.17	25.24	27.18	31.25	36.27	166.01	47.83	41.03	36.56	39.44	112.57	40.28



(a) 3 objectives

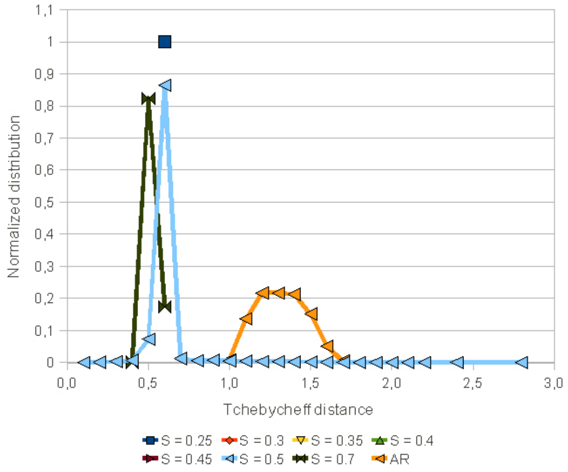
(b) 5 objectives

(c) 10 objectives

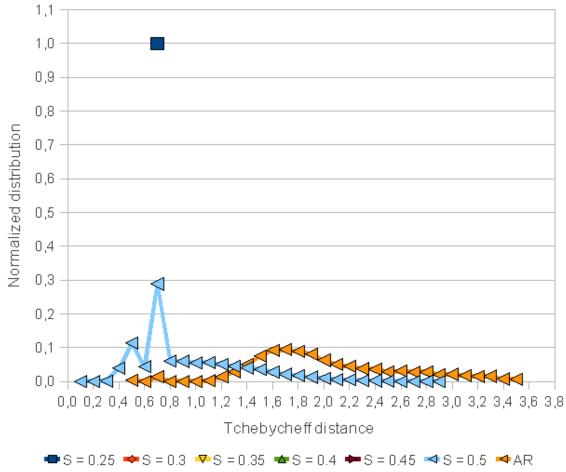
(d) 15 objectives

(e) 20 objectives

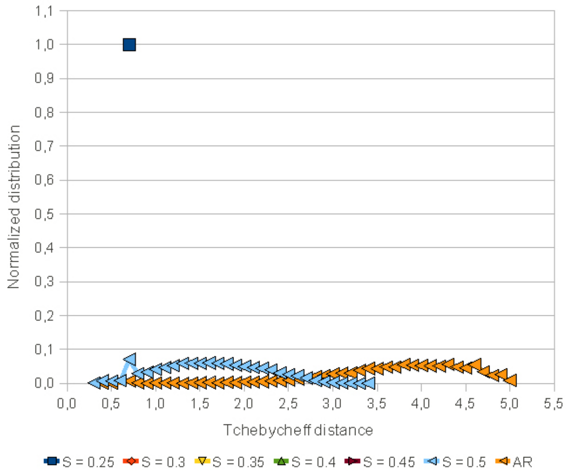
Figure. 3: Distribution of Tchebycheff distance for CDAS-SMPSO (best GD or IGD) and AR-SMPSO in DTLZ2 problem.



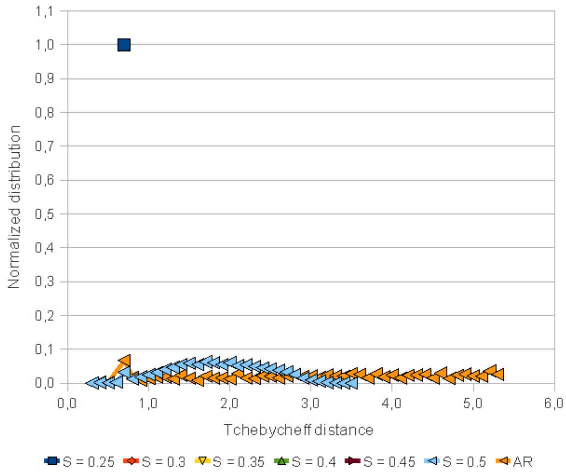
(a) 3 objectives



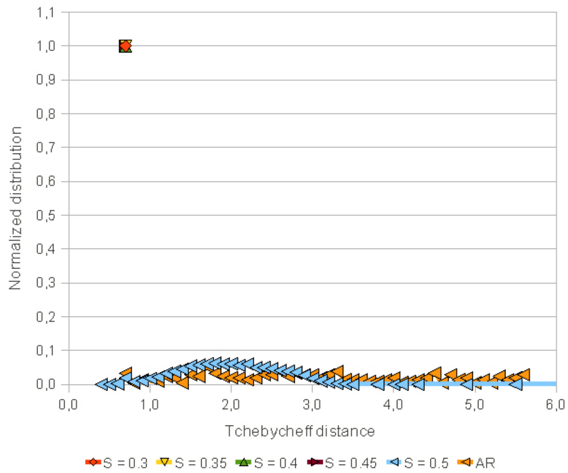
(b) 5 objectives



(c) 10 objectives



(d) 15 objectives



(e) 20 objectives

Figure. 4: Distribution of Tchebycheff distance for CDAS-SMPSO (best GD or IGD) and AR-SMPSO in DTLZ4 problem.

its solution near the knee for both problems and for all number of objectives. Examining of the quality indicators and the distribution of Tchebycheff distance, the CDAS-SMPSO obtained very good results: good convergence, diversity and its solutions were generated near the knee of the Pareto front. The AR-SMPSO produced a more diversified distribution, but far from the knee. This result was expected because, as presented in [15] for the NSGAII algorithm, the AR tends to produce extreme solutions of the Pareto front, not near the knee.

VII. Conclusion

This work presented a study of the influence of some many-objective techniques in particle swarm optimization, for many objective problems. Two different many-objective approaches were used: Control of Dominance Area of Solution and Average Ranking. These techniques were applied to a multi-objective PSO algorithm (SMPSO) that is based in cooperation of individuals, a few explored topics in literature.

A set of empirical experiments was performed to measure how CDAS and AR affect the convergence and diversity of the PSO algorithm. Besides, the algorithms were confronted to observe which technique obtained the best results in many-objective scenarios. The CDAS-SMPSO were evaluated in two different situations, using $S_i < 0.5$, i.e, selecting a subset of the Pareto Front and using $S_i > 0.5$, i.e, performing a relaxation of the Pareto dominance relation. Ten different SMPSO configurations were used: 5 with $S_i < 0.5$ and 5 with $S_i > 0.5$. The original Pareto dominance relation ($S_i = 0.5$) was also used.

The experiments were conducted with two different many-objective problems, DTLZ2 and DTLZ4, and the number of objectives were varied in five different values: 3, 5, 10, 15 and 20. Three quality indicators were used: generational distance, inverse generational distance and spacing. Also, the execution time was analyzed. Besides, the distribution of the Tchebycheff distance of the generated solutions to the knee of the true Pareto front was analyzed.

First, the best CDAS-SMPSO configurations were obtained for both problems, for each quality indicator. After, these best configurations were confronted to AR-SMPSO algorithm and the original dominance relation. In this analysis, the best results were obtained by CDAS-SMPSO, for all indicators analyzed. Besides, this algorithm executed faster than AR-SMPSO. However, this execution time is defined by the number of solutions, the smaller the number of solutions the faster is the execution time. AR-SMPSO results were outperformed by CDAS-SMPSO, but AR-SMPSO obtained better results than the original dominance relation, especially when the number of objective grows. Furthermore, the AR-SMPSO does not require any additional parameter and it executes fast independently of the problem. Through the analysis of all quality indicators and the Tchebycheff distance, it can be concluded that CDAS-SMPSO was the best technique. It generates its solutions near the knee of the Pareto front, for all problems and objectives. The AR-SMPSO produced a diversified distribution, often far from the knee. This occurs, because the AR technique prefers extreme solutions in the Pareto front.

Future works include expanding the experiments to a higher number of problems and objectives, to search for other many-objective techniques and to confront the results of PSO with other MOEAs.

References

[1] H. Aguirre and K. Tanaka. Adaptive e-ranking on many-objective problems. *Evolutionary Intelligence*, 2(4):183–206, 2009.

[2] H. Aguirre and K. Tanaka. Space partitioning evolutionary many-objective optimization: Performance analysis on mnk-landscapes. *Transactions of the Japanese Society for Artificial Intelligence*, 25(2):363–376, 2010.

[3] P. J. Bentley and J. P. Wakefield. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240. Springer-Verlag, January 1998.

[4] D. Brockhoff and E. Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervós, L. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 533–542. Springer Berlin / Heidelberg, 2006.

[5] D. Brockhoff and E. Zitzler. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation*, 17(2):135–166, 2009.

[6] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] D. W. Corne and J. D. Knowles. Techniques for highly multi-objective optimisation: some nondominated points are better than others. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780, New York, NY, USA, 2007. ACM.

[8] I. Das. On characterizing the knee of the pareto curve based on normal-boundary intersection,, volume = 18, year = 1999. *Structural Optimization*, (2-3):107–115.

[9] A. B. de Carvalho and A. Pozo. Analyzing the control of dominance area of solutions in particle swarm optimization for many-objective. In *2010 10th International Conference on Hybrid Intelligent Systems*, pages 103–108, 2010.

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

[11] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC 2002)*, pages 825–830, 2002.

[12] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[13] V. E. Gopal, M. V. N. K. Prasad, and V. Ravi. A fast and elitist multiobjective genetic algorithm: NSGA-II. *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, 2:121–136, 2010.

[14] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *CEC 2008. IEEE Congress on Evolutionary Computation*, pages 2419–2426, 2008.

[15] A. L. Jaimes and C. A. C. Coello. Study of preference relations in many-objective optimization. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO '09*, pages 611–618, 2009.

[16] A. L. Jaimes, C. A. C. Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 673–680, New York, NY, USA, 2008. ACM.
[17] A. L. Jaimes, L. V. S. Quintero, and C. A. C. Coello. Ranking methods in many-objective evolutionary algorithms. *Studies in Computational Intelligence*, 193:413–434, 2009.
[18] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE Press, 1995.
[19] I. Kokshenev and A. P. Braga. An efficient multi-objective learning algorithm for rbf neural network. *Neurocomputing*, 73(16-18):2799–2808, 2010. 10th Brazilian Symposium on Neural Networks (SBRN2008).
[20] M. Köppen and K. Yoshida. Many-objective particle swarm optimization by gradual leader selection. In *ICANNGA '07: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I*, pages 323–331, Berlin, Heidelberg, 2007. Springer-Verlag.
[21] P. Lindroth, M. Patriksson, and A.-B. Strömberg. Approximating the pareto optimal set using a reduced set of objective functions. *European Journal of Operational Research*, 207(3):1519–1534, 2010.
[22] A. Nebro, J. Durillo, J. Garcia-Nieto, C. A. C. Coello, F. Luna, and E. Alba. SMPSO: A new pso-based metaheuristic for multi-objective optimization. In *IEEE symposium on Computational intelligence in miulti-criteria decision-making, 2009. mcdm '09*, pages 66–73, 2009.
[23] V. Pareto. Manuel d"economie politique, 1927.
[24] M. Reyes-Sierra and C. A. C. Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.
[25] A. H. T. K. Sato, H. Effects of moea temporally switching pareto partial dominance on many-objective 0/1 knapsack problems. *Transactions of the Japanese Society for Artificial Intelligence*, 25(2):320–331, 2010. cited By (since 1996) 0.
[26] H. Sato, H. E. Aguirre, and K. Tanaka. *Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs*, pages 5–20. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization. Springer, Berlin, 2007.
[27] U. K. Wickramasinghe and X. Li. Using a distance metric to guide pso algorithms for many-objective optimization. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 667–674, New York, NY, USA, 2009. ACM.
[28] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.
[29] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Author Biographies

André Britto de Carvalho is a Ph.D student at the Computer Science Department on Federal University of Parana, Brazil. He received the Master degree in Computer Science at Federal University of Parana in 2009 and the BS degree in Computer Science at Federal University of Sergipe in 2006. His main interests are evolutionary algorithms metaheuristic and multiobjective optimization. Email: andrebc@inf.ufpr.br

Aurora Trinidad Ramirez Pozo is associate professor of Computer Science Department and Numerical Methods for Engineering at Federal University of Parana,Brazil, since 1997. She received a M.S. in electrical engineering from Federal University of Santa Catarina, Brazil, in 1991. She received a Ph.D. in electrical engineering from the Federal University of Santa Catarina, Brazil. Aurora's research interests are in evolutionary computation, data mining and complex problems.