

## A hybrid competent multi-swarm approach for many-objective problems

Olacir R. Castro Jr  
Computer Science's Department  
Federal University of Paraná  
Curitiba, Paraná  
olacirjr@gmail.com

Aurora Pozo  
Computer Science's Department  
Federal University of Paraná  
Curitiba, Paraná  
aurora@inf.ufpr.br

**Abstract**—Many-objective optimization problems (MaOPs) are a class of multi-objective problems that presents more than three functions to be optimized. As most Pareto based algorithms scale poorly according to the number of objectives, researchers are working on alternatives to overcome these limitations. An algorithm that has shown good results in solving MaOPs is the Iterated Multi-swarm (I-Multi) which presents a clever multi-swarm strategy to spread the solutions across different areas of the objective space while keeping a good convergence. As the I-Multi is a very recent algorithm, alternative approaches are yet to be explored. Here we investigate the use of an Estimation of Distribution Algorithm (EDA) in the multi-swarm stage of I-Multi. EDAs create a model based on the best solutions found and sample new solutions based in this model. An EDA that presents good performance is the rBOA which is a real-valued version of the Bayesian optimization algorithm. This work presents an algorithm called C-Multi consisting of a hybrid between the I-Multi and the rBOA with the aim to join the diversity strength of I-Multi and the convergence characteristic of rBOA. An experimental study is conducted using the seven well-known DTLZ test functions with 3, 5, 10, 15 and 20 objectives to evaluate the performance of the algorithms as the number of objectives scales up. The results point that the new algorithm presents superior convergence and diversity on hard problems.

**Keywords**—Particle Swarm Optimization, Many-objective, Estimation of density algorithm, Competent algorithm

### I. INTRODUCTION

Multi-objective Optimization Problems (MOPs) present the goal of optimizing two or more objective functions which are usually in conflict, resulting in a set of optimal solutions that may be found using Pareto optimality theory.

Many-objective optimization problems (MaOPs) are referred to, as a class of MOPs which presents more than three objective functions. Several studies pointed that Pareto based algorithms scale poorly in MaOPs [1], [2], [3] because of the increase in the number of non-dominated solutions which deteriorates the selection pressure.

A promising meta-heuristic to solve many-objective problems is the Multi-Objective Particle Swarm Optimization (MOPSO) which is a multi-objective version of the well-known Particle Swarm Optimization (PSO) [4], a stochastic meta-heuristic created to optimize nonlinear functions based on the movement of bird flocks looking for food.

A MOPSO algorithm that has shown good results is the I-Multi [2], which presents a clever multi-swarm strategy to enhance its performance in many-objective scenarios. I-Multi has two distinct phases: a diversity phase, where it obtains a diversified basis front and a multi-swarm phase where each sub-swarm run independently to achieve a better convergence. This two-stage idea presents very good results because it allows the search to focus precisely in the goals of enhancing the diversity and the convergence.

As the I-Multi is a very recent algorithm, alternative approaches are yet to be explored. Here is investigated a possible alternative for the second phase of the I-Multi focusing on the competent algorithms or Estimation of Distribution Algorithms (EDAs). An EDA which presents good results in the literature, as well as a good documentation is the rBOA [5], which is a modification of the Bayesian Optimization Algorithm (BOA) developed to work with real-valued variables.

This paper present a hybrid algorithm called C-Multi that join the strengths of the I-Multi algorithm of obtaining a good diversity of solutions through the use of multi-swarms to spread the solutions across the front, with the rBOA capacity of achieving good convergence by generating solutions learned from the shape of the Pareto front. With this in mind the algorithm should be able to have a better performance especially in problems presenting high difficulty, with a moderate increase in the computational cost.

The remaining of this paper is organized as follows: some background concepts are presented in Section II. Section III presents the new proposed C-Multi algorithm. An empirical study comparing the new C-Multi with I-Multi is presented in Section IV and Section V presents the conclusions.

### II. BACKGROUND

#### A. Many-objective optimization

A Multi-Objective optimization problem (MOP) is a type of problem that requires the simultaneous optimization of two or more objective functions. These objectives are usually in conflict, so these problems does not present only one optimal solution (as in single objective optimization problems), but a set of them. This set of solutions usually is found using Pareto optimality theory.

A general unconstrained MOP can be defined as optimizing  $\vec{f}(\vec{x}) = (f_1(\vec{x}), \dots, f_m(\vec{x}))$ , where  $\vec{x} \in \Omega$  is an  $n$ -dimensional decision variable vector  $\vec{x} = (x_1, \dots, x_n)$  from a universe  $\Omega$ , and  $m$  is the number of objective functions.

An objective vector  $\vec{f}(\vec{x})$  dominates a vector  $\vec{f}(\vec{y})$ , denoted by  $\vec{f}(\vec{x}) \preceq \vec{f}(\vec{y})$  (in case of minimization) if and only if  $\vec{f}(\vec{x})$  is partially less than  $\vec{f}(\vec{y})$  i.e.,  $\forall i \in \{1, \dots, m\}, f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1, \dots, m\} : f_i(\vec{x}) < f_i(\vec{y})$ .

A vector  $\vec{f}(\vec{x})$  is non-dominated if there is no  $\vec{f}(\vec{y})$  that dominates  $\vec{f}(\vec{x})$ . If  $\vec{f}(\vec{x})$  is non-dominated,  $\vec{x}$  is Pareto optimal. The set of Pareto optimal solutions is called Pareto optimal set, and the image of these solutions in the objective space is called Pareto front [6].

Many-Objective Optimization Problems (MaOPs) are a type of MOPs with the characteristic of presenting more than three objective functions to be optimized simultaneously. Several studies have indicated that Pareto based algorithms scale poorly in MaOPs [1], [2], [3]. The main reason for this is the number of non-dominated solutions which increases greatly with the number of objectives, as consequence, the search ability is deteriorated because it is not possible to impose preferences for selection purposes.

## B. MOPSO

Particle Swarm Optimization (PSO) [4] is a stochastic meta-heuristic created to optimize nonlinear functions based on the movement of bird flocks looking for food. In this method a swarm (population) of particles (solutions) moves across the search space (evolve) guided by personal and social leaders.

To expand the PSO to solve multi-objective problems, and create a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm, some modifications are needed. The first of them is the creation of an external archive (repository) to store the better (non-dominated) solutions found so far, another modification is in the leader selection scheme, which has to choose from a set of equally good leaders according to some criterion. As the number of non-dominated solutions may become very large, an archiving method is needed to prune the repository and keep only a predefined number of solutions, discarding some non-dominated solutions according to its criterion.

A MOPSO which has shown very good results in the literature is the Speed-constrained Multi-objective PSO (SMPSO) [7]. It was noted that in some conditions the velocity of the particles in a MOPSO can become too high, generating erratic movements towards the limits of the decision space, to avoid such situations SMPSO presents a velocity constriction mechanism based on a factor  $\chi$  that varies based on the values of the influence coefficients of personal and global leaders ( $C_1$  and  $C_2$  respectively). In SMPSO the (global) leader selection method uses a binary tournament based in the Crowding Distance metric from [8], and the archiving strategy also uses the Crowding Distance.

As the global leaders are selected from the repository, its size and content (managed by the archiving method) have a great impact on the results, especially in MaOPs, where the number of non-dominated solutions quickly increases. In [3] several archiving methods are compared in the optimization of MaOPs. The methods that do not limit the number of solutions in the Pareto set presented the best results, and among the methods that limit the size of the archive, the Ideal and MGA [9] stands out.

Multi-level Grid Archiving (MGA) [9] is an archiver that keeps the repository with a limited number of solutions, when the repository becomes full, it divides the objective space into boxes and every solution in the archive has a box index. Then it is observed the dominance relation between the boxes. If the new solution to be added belongs to one of the dominated boxes, it is not included in the repository otherwise one of the solutions that have dominated box indexes is removed randomly. If there is no dominance relation between the boxes, the objective space is split again into smaller boxes until a dominated box is found.

In the Ideal archiver, when the repository becomes full it firstly obtains the ideal point of the repository. The ideal point is a vector with the best values of a set for every objective. Then, it calculates the Euclidean distance from each solution in the archive to the ideal point. The solution with the highest distance is removed.

## C. I-Multi

Recently a new MOPSO called I-Multi [2] was proposed. I-Multi is a MOPSO designed to deal with many-objective problems and uses multiple swarms to cover different areas of the objective space, showing good results in the literature due to this strategy. Its search procedure can be divided in two phases: diversity and multi-swarm searches. In the first phase, a diversity search is performed using SMPSO [7] with the MGA archiver [9] to generate a set of well-distributed non-dominated solutions (basis front) for multi-swarm initialization. Multi-swarm search begins using the basis front to assign a seed to each sub-swarm and around each seed (within a specified search region), a population of solutions is randomly generated to form a sub-swarm.

During a predefined number of iterations each sub-swarm run independently, using the SMPSO with the Ideal archiver [3] to enhance its convergence. After that, the repository of each sub-swarm is integrated to the basis front so, only the non-dominated solutions regarding all repositories are kept. At the end of this process, the basis front is split in sub-swarms as before. This process of joining and splitting the fronts is called split iteration, and is repeated a predefined number of times. This process enables an indirect communication between the sub-swarms.

In [2] three methods are presented to define the seed and the archive of each sub-swarm. In this work is used the I-Multi centroid, where the basis front is split in clustered

groups (using a K-Means algorithm in the decision space) and the centroid of each group is chosen as seed for each sub-swarm. The solutions of each cluster are stored in each repository. The number of clusters is defined as the same number of swarms. In this strategy, the archive is updated with all solutions clustered in the same cluster of the seed.

#### D. Estimation of distribution algorithms

In spite of the success of the Evolutionary Algorithms (EAs), they face difficulties in some real-world problems presenting properties like non-linearity, ill-conditioning and deception mainly because they do not properly consider the dependencies and relationships between decision variables and are not able to thoroughly exploit the information obtained so far.

Probabilistic modeling offers a systematic way of acquiring problem regularities, and therefore can help in problem solving by replacing the traditional genetic operators in the process of generating new solutions by estimating a probabilistic model based on the statistic collected from the set of solutions and then sampling this probabilistic model.

Therefore problem regularities encoded in the probabilistic model are used to generate new solutions to overcome the limitations of traditional EAs. This incorporation of probabilistic modeling in EAs has led to a new paradigm known as Estimation of Distribution Algorithms (EDAs) or competent EAs [10].

General EDAs iterate three steps until some termination criterion is satisfied: select good solutions from a population, estimate the probability distribution from the selected individuals (learn a model) and generate new solutions (sample) from the estimated distributions (model).

In spite of similar behavior patterns, EDAs can be characterized by its methods of learning a probabilistic model, which consist of two tasks: learning the structure of a probabilistic model, which defines conditional dependencies and independencies (model selection), and estimating the conditional probability distributions with regard to the found structure (model fitting) [5].

The EDA used in this work is the real-coded Bayesian optimization algorithm (rBOA) [5], which was chosen due to its good results and extensive documentation available in [5]. rBOA is a continuous, real-coded version of the Bayesian optimization algorithm (BOA) proposed to bring the power of the discrete BOA to the area of real-valued optimization.

rBOA model selection obtains a Bayesian factorization, which is represented by a directed acyclic graph called Bayesian factorization graph where the nodes and edges identify the corresponding variables and its conditional dependencies respectively. To learn the structure of a probabilistic model (model selection), two basic factors are used: a scoring metric and a search procedure. The scoring metric assess the quality of the structure of the Bayesian factorization graph and the search procedure efficiently

traverses the space of all feasible structures for finding the best one with regard to a given scoring metric. BOA and rBOA employ a Bayesian Information Criterion (BIC) as the scoring metric and an incremental greedy algorithm as the search procedure [5].

In rBOA model fitting a set consisting of a node and its parents in the Bayesian factorization graph represents a component sub-problem of decomposable problems. For efficiency, maximal connected sub-graphs of a Bayesian factorization graph are considered sub-problems and must be independently fitted. Mixture models are employed as an efficient tool for the purpose and its aim is twofold: comprehending the type of dependency between variables and traversing the search space effectively. Each mixture component can model the linearity of the variables, thus they can approximate any type of dependency by a combination of piecewise linear interaction models [5].

### III. COMPETENT MULTI-SWARM

This section presents the new Competent Multi-swarm algorithm (C-Multi) which is created to combine the high convergence of the EDAs with the good results presented by the I-Multi in many-objective scenarios.

As in I-Multi, the optimization process of C-Multi is carried in two phases: diversity and multi-swarm. In the diversity phase, a traditional SMPSO [7] with MGA archiver [9] is used to obtain a diverse set of non-dominated solutions (basis front) to be used in the next phase.

The multi-swarm phase begins by splitting the basis front in  $NS$  fronts, each one with its closest non-dominated solutions and a seed (centroid of the cluster). This seed is used to define the bounds to truncate the solutions (if necessary) to make sure each sub-swarm will explore only a limited region to avoid them to overlap.

During a predefined number of runs, the non-dominated solutions in each front are used to learn a model and then sample this model to create a new population (or replace a previous one) thus forming a sub-swarm. Then the population of each sub-swarm is added to the front meeting the criteria of the Ideal archiver [3], completing the iteration.

After a predefined number of iterations, the multi-swarm execution is interrupted to start a split iteration, in which the fronts of all the sub-swarms are merged into a single basis front (only the non-dominated solutions concerning all fronts are kept) that is split as before, and the multi-swarm phase restarts. The split iterations are done a predefined number of times to allow an indirect communication between the sub-swarms. The new C-Multi pseudocode is presented through Algorithm 1.

In Algorithm 1, firstly the basis front ( $F_b$ ) is generated using the SMPSO with MGA archiver. Next the multi-swarm stage begins, for  $SI$  times  $F_b$  is split (using a K-Means algorithm in the decision space) in  $NS$  subsets

$\vec{F}=(F_1, \dots, F_{NS})$ , with each subset ( $F_k$ ) having a set of non-dominated solutions and a seed.

---

**Algorithm 1: C-Multi**

---

```

//Phase1: Diversity search
 $F_b$ =Run-MGA-SMPSO()
//Phase 2: Multi-swarm search
for  $s=1$  to  $SI$  do
     $\vec{F}$ =SplitFront( $F_b$ )
    for  $k=1$  to  $NS$  do
        for  $i=1$  to  $It$  do
             $P_k$ =rBOA( $F_k$ )
            Truncate( $P_k$ )
            Evaluate( $P_k$ )
             $F_k$ =IdealArchiving( $P_k$ )
        end for
    end for
     $F_b$ =Non-dominated( $\vec{F}$ )
end for
return  $F_b$ 

```

---

In the next step a model is learned from  $F_k$  and a new population ( $P_k$ ) is sampled from this model and its decision vector is truncated (if necessary) to stay within the search region around the seed. The new population is then evaluated through the objective functions and incorporated in  $F_k$  using the Ideal archiver to meet the non-domination criteria and keep the number of solutions limited. The ideal point of the archive is calculated independently for each sub-swarm. After each sub-swarm execute for ( $It$ ) iterations,  $F_b$  is updated and split again into  $NS$  sets and the process repeats  $SI$  times. When  $SI$  split iterations are done the results from the updated archive are returned.

The structure of C-Multi is basically the same as the I-Multi, however in the multi-swarm search, instead of using the SMPSO, the rBOA learning method is used to create a model and sample new particles. This replacement of update method is used to increase the convergence capacity in this stage of the search, causing a moderate increase in the computational cost.

#### IV. EMPIRICAL STUDY

This session presents the empirical study conducted to compare the performances of the new C-Multi and its base algorithm I-Multi in many-objective scenarios, to evaluate how both approaches behaves, especially as the number of objectives scale up.

The experimental setup is described in Section IV-A, and the results and its analysis are presented in Section IV-B.

##### A. Experimental setup

The parameters used in this study follows the same used in [2], with the inertia factor ( $\omega$ ) randomly selected in [0,0.8], and  $C_1, C_2$  in the interval [1.5,2.5]. The initial phase uses the MGA as archiver with 100 generations, population of 100 individuals and archive size of 200 solutions.

In the multi-swarm search each sub-swarm uses the Ideal archiving method and run for 100 iterations. The specific parameters of the multi-swarm phase were chosen according to the results shown in [2] always choosing the parameters with better performance. The search region size was set to decrease from 0.5 to 0.1 along the search, the number of split iterations was set to 5, the number of swarms is set to 30, the population per swarm is set to 25 and the archive size of each swarm was kept in 200. The learning parameters were set according to [5], with leader threshold of 0.3 and Bayesian information criterion  $\lambda$  set to 0.5.

In this work the entire DTLZ [11] family of well-known multi-objective optimization problems were used (DTLZ1 - DTLZ7). These problems can scale both in number of objectives and in number of decision variables, also the true Pareto optimal front is known. Both the algorithms were tested using all the problems with 3, 5, 10, 15 and 20 objectives to verify how they perform as the number of objectives scales up.

To assess the quality of the Pareto fronts generated by each algorithm two quality indicators are used: The modifications of the well-known metrics Generational Distance (GD) and Inverted Generational Distance (IGD) known as  $GD_p$  and  $IGD_p$  respectively [12], these modifications make both metrics more fair and Pareto compliant, however do not change its main characteristics.

$GD_p$  measures the average distance from the obtained front to the real front (discretized), this way is possible to assess the convergence of the obtained solutions towards the real front. A value of zero means that all the solutions are optimal i.e. reached the true front.

The  $IGD_p$  on the other hand measures the average distance from the real front to the obtained solutions. Its values also represents in a certain level the convergence to the real front, but mainly represent the diversity of the solutions over the entire front. A value of zero means that the obtained front and the real front are equal.

It is important to observe both  $GD_p$  and  $IGD_p$  metrics together to have a perspective on what algorithm presents better performance in convergence to the real front and in diversity of the solutions as these two characteristics are often conflicting, but our main goal is to enhance both.

The results obtained with these metrics in 30 independent runs of the algorithms for every number of objectives are compared using the Friedman statistical test [13] at 5% significance level, the post-test indicates if there are statistical differences between the results and the mean of the values is used to determine the algorithm with the better values.

##### B. Results and discussion

This section presents the results obtained with the I-Multi and the new C-Multi algorithm optimizing the seven DTLZ problems. The results of both are compared in terms of

Table I  
GD<sub>p</sub> AND IGD<sub>p</sub> RESULTS

Obj	Algorithms	DTLZ1		DTLZ2		DTLZ3		DTLZ4	
		GD <sub>p</sub>	IGD <sub>p</sub>	GD <sub>p</sub>	IGD <sub>p</sub>	GD <sub>p</sub>	IGD <sub>p</sub>	GD <sub>p</sub>	IGD <sub>p</sub>
3	I-Multi	1.25E+02	9.96E+00	1.13E-01	5.00E-02	6.85E+02	1.94E+02	2.56E-01	3.96E-02
	C-Multi	1.50E+02	1.56E+01	2.12E-01	7.78E-02	7.51E+02	2.41E+02	3.54E-01	5.83E-02
5	I-Multi	1.58E+02	1.61E+01	4.28E-01	1.83E-01	8.34E+02	2.73E+02	6.77E-01	1.15E-01
	C-Multi	1.57E+02	1.49E+01	5.60E-01	2.34E-01	8.22E+02	2.74E+02	6.70E-01	1.45E-01
10	I-Multi	1.66E+02	1.46E+01	6.22E-01	2.44E-01	9.08E+02	2.52E+02	8.25E-01	2.01E-01
	C-Multi	1.51E+02	1.41E+01	7.11E-01	3.02E-01	8.64E+02	2.78E+02	8.25E-01	2.21E-01
15	I-Multi	1.66E+02	1.42E+01	6.58E-01	2.49E-01	9.14E+02	2.70E+02	8.58E-01	2.47E-01
	C-Multi	1.51E+02	1.46E+01	7.34E-01	3.09E-01	8.79E+02	2.84E+02	8.63E-01	2.71E-01
20	I-Multi	1.68E+02	1.35E+01	6.74E-01	2.53E-01	9.20E+02	2.69E+02	8.65E-01	2.95E-01
	C-Multi	1.55E+02	1.60E+01	7.42E-01	3.19E-01	8.87E+02	2.90E+02	8.71E-01	3.05E-01

  

Obj	Algorithms	DTLZ5		DTLZ6		DTLZ7	
		GD <sub>p</sub>	IGD <sub>p</sub>	GD <sub>p</sub>	IGD <sub>p</sub>	GD <sub>p</sub>	IGD <sub>p</sub>
3	I-Multi	3.61E-02	4.20E-03	8.46E+00	7.50E+00	4.78E+00	1.33E+00
	C-Multi	7.27E-02	8.20E-03	8.53E+00	7.57E+00	6.61E+00	1.86E+00
5	I-Multi	8.42E-01	4.24E-02	9.09E+00	7.77E+00	1.54E+01	4.82E+00
	C-Multi	7.79E-01	5.79E-02	9.11E+00	7.80E+00	1.65E+01	5.62E+00
10	I-Multi	8.87E-01	6.75E-02	9.15E+00	7.83E+00	3.93E+01	1.35E+01
	C-Multi	8.61E-01	7.44E-02	9.16E+00	7.96E+00	3.97E+01	1.48E+01
15	I-Multi	8.79E-01	8.03E-02	9.15E+00	7.82E+00	6.16E+01	2.25E+01
	C-Multi	8.73E-01	8.90E-02	9.16E+00	7.95E+00	6.16E+01	2.54E+01
20	I-Multi	8.83E-01	8.34E-02	9.15E+00	7.89E+00	8.30E+01	2.88E+01
	C-Multi	8.74E-01	9.30E-02	9.16E+00	7.94E+00	8.33E+01	3.04E+01

convergence to the real front and diversity of the solutions obtained as the number of objectives scales up.

A study comparing the computational times of the algorithms was not included in this work due to lack of space, but in our experiments, the C-Multi algorithm presented an execution time moderately higher than its base algorithm.

The results are presented through Table I, where the average GD<sub>p</sub> and IGD<sub>p</sub> of 30 runs are presented for both algorithms in 3, 5, 10, 15 and 20 objectives on each problem. The best results (and the statistically equivalent according to the Friedman test) are highlighted. Charts were not included in this work due to lack of space.

According to Table I, for three objectives the I-Multi had the better results both in convergence and diversity for all problems, with C-Multi sharing the best result only in diversity of the problem DTLZ6. This advantage is achieved due to the small number of objectives, which does not decrease the selection pressure towards the front, then the particles are able to find better solutions. With more objectives, the learning process of rBOA has advantage because it generates solutions close to the better front found so far.

From five to twenty objectives, the behavior of each algorithm is similar even changing the number of objectives, so for now on, the analysis will be conducted by problem from five objectives onwards.

In DTLZ1, which is a very difficult problem due to its large objective space, the C-Multi had the best convergence for ten or more objectives, however its diversity results had no statistically relevant difference. This occurs because

with a large objective space, the particles in SMPSO have difficulties to remain close to the front, however the C-Multi learns the model from the better front found so far and then generate more solutions close to it.

The DTLZ2 is a problem where MOPSOs generally reach good results as this problem do not impose much challenge to the algorithms, in this function the I-Multi performed better both in convergence and diversity in all cases.

DTLZ3 is a modification of DTLZ2 with the difficulties of DTLZ1 i.e. a large objective space. In this problem the C-Multi presented better convergence, and showed almost equal results to the I-Multi in diversity. As in DTLZ1, these better results are due to the capacity of C-Multi of generating solutions close to the best front found so far.

In DTLZ4, which challenge the capacity of the optimizers to obtain a diversified solution set, the I-Multi had a better overall performance. I-Multi had the better performance in every number of objectives, sharing the position with C-Multi in convergence for five and ten objectives, and in the diversity for twenty objectives. Although being a hard problem, DTLZ4 does not impose much convergence difficulty, which is where the C-Multi stands out.

The DTLZ5 is an easy problem which tests the ability of an optimizer to converge to a curve. In this problem the C-Multi had better results in convergence for five and ten and shared the best position with I-Multi in fifteen and twenty objectives, however the I-Multi had better results in terms of diversity for all objectives. This occurs because the C-Multi started with a set of solutions very close to the true front,

for this reason generated solutions also very close to the front, while the I-Multi spread more the solutions through the front, sacrificing some convergence in favor of diversity.

DTLZ6 is a harder version of DTLZ5 (larger objective space). In this problem the I-Multi had the better results both in convergence and diversity, sharing its best position in diversity with C-Multi for three and twenty objectives. In this case, opposite to DTLZ5 the initial set of solutions used in learning of C-Multi was not so good, as this problem is harder, resulting in worse results.

In the DTLZ7, which present a disconnected Pareto front, the I-Multi also had the advantage both in convergence and diversity due to the confusion caused to the learning algorithm of C-Multi in the disconnected environment.

## V. CONCLUSION

This paper presented the new C-Multi algorithm designed to combine the strong points of both the I-Multi and the rBOA algorithms with a moderate increase in the computational cost. The I-Multi is able to provide a well-spread set of solutions due to the multi-swarm strategy while achieving convergence especially in the easier problems by using an archiver which privileges convergence. The rBOA is able to generate solutions close to the true Pareto front by creating a model that represents the best solutions found so far and modeling the relationships between its variables, then sampling new solutions from this model.

An experimental study was conducted comparing the performance of the C-Multi algorithm with its base algorithm I-Multi. The seven well-known DTLZ problems were used with 3, 5, 10, 15 and 20 objectives to evaluate the scalability of both algorithms. Statistical tests were performed to confirm if the difference in the results were relevant.

From the results presented is possible to conclude that the new C-Multi had better behavior in the harder problems with large objective space and many local optima as the DTLZ1 and DTLZ3 [11], however the I-Multi had the better overall performance, achieving good results both in diversity and convergence in most of the problems.

These better results found by the C-Multi can be explained by its learning characteristic, in which the sampled solutions are not much far away from the better solutions found in each iteration, increasing the exploitation capacity, however the exploration might be decreased.

These results encourage further study of the hybridization between EDAs and MOPSOs to take advantage of their different capabilities in order to create a more adaptive algorithm to explore the diverse scenarios that can be found in real-world many-objective optimization.

Future works include comparing the hybridization of I-Multi with different learning algorithms and conducting a more detailed study to calibrate the learning parameters for these new applications.

## REFERENCES

- [1] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima, "Behavior of emo algorithms on many-objective optimization problems with correlated objectives," in *CEC 2011. IEEE Congress on Evolutionary Computation*, June 2011.
- [2] A. Britto, S. Mostaghim, and A. Pozo, "Iterated multi-swarm: A multi-swarm algorithm based on archiving methods," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 583–590.
- [3] A. Britto and A. Pozo, "Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, June 2012, pp. 1–8.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, nov/dec 1995, pp. 1942–1948 vol.4.
- [5] C. W. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice (Studies in Computational Intelligence)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [6] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [7] A. Nebro, J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *Computational intelligence in multi-criteria decision-making, 2009.*, Apr. 2009.
- [8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, ser. PPSN VI. London, UK: Springer-Verlag, 2000, pp. 849–858.
- [9] M. Laumanns and R. Zenklusen, "Stochastic convergence of random search methods to fixed size pareto front approximations," *European Journal of Operational Research*, vol. 213, no. 2, pp. 414 – 421, 2011.
- [10] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *Journal of Heuristics*, vol. 18, no. 5, pp. 795–819, 2012.
- [11] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, vol. 1, may 2002, pp. 825 –830.
- [12] O. Schutze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 4, pp. 504–522, Aug 2012.
- [13] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.