Differential Evolution variants combined in a Hybrid Dynamic Island Model

Grasiele Regina Duarte
Civil Engineering Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
grasiele.duarte@coc.ufrj.br

Beatriz Souza Leite Pires de Lima
Civil Engineering Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
bia@coc.ufrj.br

***Abstract*—Several Variants of the Differential Evolution (DE) algorithm have been among the winners of competitions on bound constrained optimization problems in distinct editions of the Congress on Evolutionary Computation (CEC). This paper compares some of those variants presented in the last five years. Those algorithms are also combined in two adaptive dynamic Island Models (IM), which adjust the topology according to the algorithms applied in their islands. The Island Models are able to adaptively direct more solutions to the islands with more qualified algorithms to solve the problem. The results allow to verify whether the combination of the recently proposed DE variants in an adaptive dynamic structure is better than using each one individually.**

***Index Terms*—Differential Evolution, Competition of Evolutionary Algorithms, Adaptive Dynamic Island Model.**

I. Introduction

Differential Evolution (DE) is a simple and efficient Evolutionary Algorithm (EA). Since it was proposed, DE attracted attention of many researchers. Many variants and improvements for DE were proposed over the years.

Periodically, the good performance of DE and its variants is reinforced in the competition of EAs, to solve bound constrained optimization problems, promoted in the Congress on Evolutionary Computation (CEC). At each edition of CEC, DE variants which participate of the competition, generally take position among the winners. However, the benchmark problems used in the competition are usually changed at each CEC edition and winner algorithms of previous editions are not necessarily compared with competitors of the current edition.

It is important to know the impacts promoted to DE by good variants proposed over the years. An alternative to evaluate those algorithms is to compare the quality of the solutions produced by them, as generally done in CEC competitions. However, it is also interesting to check which features of the base algorithm are changed by the new variants. It is pertinent to verify the reasons for the improvements promoted to DE performance.

This paper performs an additional evaluation of some variants of DE, proposed in the last 5 years, included among the winners of CEC competitions on bound constrained optimization problems. Those algorithms are also incorporated in two adaptive dynamic Island Models (IM). In this way, it is possible to evaluate if combining some DE variants is better than just choosing and using one of them individually. The algorithms are compared under the same benchmark problems.

II. Differential Evolution

The DE and some of its variants were proposed in [1] and became one of the most studied and used EAs in the literature [2]–[4]. In DE, the set of candidate solutions (population) is composed by NP D-dimensional vectors, where D is the dimension of the problem and NP is defined by the user. At each DE iteration, three operations are applied to each candidate solution: mutation, crossover and selection. In mutation, for each vector x_i, $i = 1, 2, ..., NP$ is produced a mutant vector v_i given by

$$v_i = x_{r1} + F \times (x_{r2} - x_{r3}), i = 1, 2, ..., NP, \qquad (1)$$

where $r1$, $r2$ and $r3 \in \{1, 2, ..., NP\}$ are indexes of vectors randomly chosen, mutually different and also different from i, $F \in (0, 2]$ is a DE parameter, whose value is defined by the user [1].

In crossover, the solutions v_i and x_i are combined, producing the solution u_i as

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rand}(j) \leq CR \text{ or } j = \text{rand}(i) \\ x_{i,j}, & \text{if } \text{rand}(j) > CR \text{ and } j \neq \text{rand}(i) \end{cases}, \qquad (2)$$

where $\text{rand}(j)$ is the j-th random real value $\in [0, 1]$, $\text{rand}(i)$ is a random integer value $\in \{1, 2, ..., D\}$. In (2), $CR \in [0, 1]$ is another parameter of DE whose value is defined by the user [1].

The selection defines which solution between u_i and x_i will compose the population. Thus, the values of objective function $f(\cdot)$ of these solutions are compared in a greedy criterion. For a minimization problem, if $f(u_i) < f(x_i)$, x_i will be replaced by u_i, otherwise x_i will be kept in population [1].

In [1], it was also proposed different alternatives to implement the DE and a scheme to name such variants. In this way, an instance of DE is identified as DE$/x/y/z$, where x is the strategy adopted to define the vectors involved in mutation, y is the number of difference operations in mutation and z is the crossover scheme.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

A. Recent variants of Differential Evolution

As mentioned above, the DE has three parameters to be defined by the user: NP, F and CR. Evidently, its performance depends on the adjustment of these parameters.

Over the years some studies have proposed methodologies for dynamic and adaptive adjustment of DE parameters, mainly F and CR, as can be exemplified in [5], [6]. Particularly, in [7] it was proposed the methodology which resulted in the algorithm called JADE.

In JADE, each vector x_i has its own parameters F_i and CR_i and they are adjusted at each iteration. The adjustment of each F_i and CR_i, $i = 1, 2, ..., NP$, is based on the mean values defined for these parameters that improved solutions in previous iterations. JADE also applies a new mutation known as current-to-p-best.

The JADE was the basis for the proposal of new optimization algorithms such as those described in [8]–[12]. An important algorithm based in JADE, called SHADE, was proposed in [10].

The main difference between JADE and SHADE is that in SHADE the adjustment of each F_i and CR_i is based in two sets of values corresponding to weighted means of F_i and CR_i which improved solutions in previous iterations. Additionally, in SHADE, the parameters inherited from JADE were individualized for each candidate solution, such as the parameter p_i, used in mutation operation, adjusted randomly in a specific domain at each iteration.

SHADE performed well when compared to other DE based algorithms in the edition 2013 of CEC competition [13]. Since then, SHADE has been the basis for proposal of new algorithms that has also taken place among the winners of CEC competitions in the last years. The following paragraphs present a brief description of some of those algorithms.

In [14], the algorithm called L-SHADE, winner of CEC competition in 2014, was proposed. The main innovation of L-SHADE for SHADE was the linear reduction of the population as a function of the number of fitness evaluations.

Then, in 2015, the algorithms called SPS-L-SHADE-EIG [15] and DEsPA [16] were proposed, being respectively the winner and second place in the 2015 CEC competition. The SPS-L-SHADE-EIG [15] applies a mutation based on eigenvectors of the covariance matrix of the population. In addition, the choice of parent vectors for mutation and crossover operations favors recently improved solutions. DEsPA [16] seeks to manage the greediness in mutation and the adjustment of DE parameters in a optimized way. The main innovation of DEsPA, regarding L-SHADE, was the variation in the population size. In DEsPA, the population size increases, remains stable and is reduced in three distinct phases [16].

In [17], the algorithm called CCLSHADE was proposed and won the second place in CEC Learning-Based competition in 2016. In CCLSHADE, the evolution of candidate solutions occurs cooperatively. Mutation and crossover operations are applied separately in distinct groups of variables, each one formed by a set of non-separable problem variables.

The algorithm called jSO, second place in the 2017 CEC competition, was proposed in [18] as an extension of algorithm iL-SHADE [19]. The jSO, like iL-SHADE, applies a strategy to control the values assigned to parameters F_i and CR_i, especially in the early stage of the evolutionary process. High values for F_i and low values for CR_i are not allowed. In addition, the mutation greediness is manipulated along the fitness evaluations. The algorithm jSO also proposed to weight an occurrence of F_i in mutation, according to the stage of the evolutionary process [18].

The second place in the 2018 CEC competition was the algorithm called LSHADE-RSP, proposed in [20]. The LSHADE-RSP uses the mutation of jSO with an extension, where the random solutions have a choice probability according to its position in a ranking based on the quality of the solutions. LSHADE-RSP, as jSO, also controls the values of F_i and CR_i for each phase of the evolution.

III. Adaptive Dynamic Island Model

Island Model (IM) is an alternative to implement EAs to run in parallel computational environments. In IM [21], the population is partitioned into subsets called islands, which evolves in parallel by its own algorithm. Along iterations of the algorithm, the islands exchange solutions by a process called migration, using the topology connections.

It is possible to implement the IM in a hybrid way. In this case, a different algorithm is applied in each island. Other decisions to implement the IM are basically [22], [23]:

- *Number of islands*: Number of algorithms or sub-populations.
- *Migration topology*: Defines how the islands are connected, meaning the available paths for migration. Also, each connection can be uni- or bi-directional and the topology can be static or dynamic [23]–[25].
- *Migration rate*: Value used to define how many candidate solutions will migrate from each island.
- *Migration frequency*: Defines the number of migration that will be performed in IM.

The migration also depends on the *migration policy* defined by the user. In this sense, the user should decide, for example, how the migrant solutions are chosen and what island will be the destination for each one. Besides, the migration can be *synchronous* or *asynchronous*, *point to point* or *broadcast* [24], [26].

A. Adaptive Dynamic Island Model

In [27], it was proposed a dynamic hybrid IM identified in this work as D-IM. Initially, the topology of D-IM is fully connected, composed by weighted uni-directional connections. Along migrations, the weights of the connections are dynamically adjusted in $[0, 1]$ according to the attractiveness of the destination island to the source island. In D-IM, the attractiveness of an island to another is calculated according to informations of its population, provided by its algorithm.

In D-IM, at each migration, the solutions are actually moved from the source island to the destination island. In this case, the number of solutions directed to each island may be different. The idea is that islands with algorithms more qualified to solve the problem maintain a larger number of candidate solutions to evolve.

In [28], it was proposed a new strategy to evaluate the attractiveness of an island to another in D-IM. By the strategy proposed in [27], an island becomes more attractive if its algorithm has fast convergence. On the other hand, by the strategy proposed in [28], islands which produce better solutions become more attractive.

In [28], it was found that by the strategy proposed in [27] for topology adjustment, islands with exploratory algorithms become more attractive in D-IM. Differently, by the strategy proposed in [28], islands with intensifying algorithms attract more solutions to their populations. In this case, according to the strategy used, the topology adjustment in D-IM can also indicate the nature of the algorithms applied in its islands.

IV. Experiments

For the evaluation of EAs considered in this work, it was used the set of problems proposed in [29] for the competition on bound constrained optimization problems in edition 2015 of CEC. In that year, it was accepted that the competitors algorithms had their parameters adjusted differently to solve each problem, which justifies the term learning-based in the competition name [29]. The benchmark is composed by 15 minimization problems, identified in this work as Fi, where $i = 1, 2, ..., 15$. In [29], the problems were divided into 4 distinct groups. Problems $F1$ and $F2$ are Unimodal Functions, while problems $F3$, $F4$ and $F5$ are Simple Multimodal Functions. Problems $F6$, $F7$ and $F8$ are Hybrid Functions. They combine problems like the last seven problems, classified as Composition Functions [29]. In this work, dimensions $D = 10$ and $D = 30$ were considered for the evaluation of the algorithms.

The DE based algorithms compared in this work were winner or second place in CEC competitions in the last 5 years: L-SHADE [14], SPS-L-SHADE-EIG [15], DEsPA [16], CCLSHADE [17], jSO [18] and LSHADE-RSP [20], briefly presented in Section II-A. For comparison purposes, two configurations of original DE were considered and the algorithms JADE [7] and SHADE [10], predecessors of most DE based winners in the last CEC competitions.

Regarding the parameters of each algorithm, the values recommended by the authors in their respective works were used. Regarding the two configurations of DE, the following settings were used:

- DE-1: (i) $Variant = DE/best/1/bin$, (ii) $F = 0.5$, (iii) $CR = 0.9$, (iv) $NP = 50$.
- DE-2: (i) $Variant = DE/rand/1/bin$, (ii) $F = 0.5$, (iii) $CR = 0.9$, (iv) $NP = 50$.

The evaluated algorithms were also combined in the D-IM described in Section III, under both strategies proposed in [27] and [28] for topology adjustments. In this work, the D-IMs were identified as DIM-1 and DIM-2 according to the strategies proposed in [27] and [28], respectively.

Regarding the parameters of DIM-1 and DIM-2, they were defined as follows [27], [28]: (i) Migration rate: 10%; (ii) Migration frequency: 50; (iii) $M = 5$; (iv) $\theta = 0.05$ (v) $POP = 200$ for $D = 10$ and $POP = 300$ for $D = 30$, initially divided equally between islands, where $POP =$ population size.

The algorithms that change the population size along their run had such operations disabled when combined in DIM-1 and DIM-2. In this case, due to the fact that the algorithms are applied in islands of DIM-1 and DIM-2, their population size are now controlled by the the migration processes of the models.

For each problem, DIM-1, DIM-2 and each individual algorithm were performed using 30 independent runs. The Maximum Number of Function Evaluations was set as $MFE = 10000 \times D$.

Usually, the performance difference between EAs is verified by comparing the objective function values of solutions obtained for each problem in the runs. Generally, the comparison is also guided by statistical metrics as mean, median and standard deviation defined on these values.

In this work, the performance of the algorithms regarding the quality of solutions was compared by the systematic methodology called Performance Profile, proposed in [30]. This technique is applicable in evaluations where there are a set of algorithms and a set of problems. The analysis is based on a performance measurement of the algorithms defined by the user.

For a brief description of the Performance Profile technique, suppose a set S with n_s algorithms and a set P with n_p problems. Also suppose the computational time of each algorithm as the performance measure. Thus, for each problem $p \in P$ and $s \in S$ is defined the value

$$t_{p,s} = \text{computational time required to solve } p \text{ with } s. \quad (3)$$

Then, for each problem p, the performance of each algorithm s is compared to the best one in S by the metric $r_{p,s}$ given by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}, \quad (4)$$

where $\min\{t_{p,s} : s \in S\}$ is the $t_{p,s}$ of the algorithm in S that solved the problem p in shortest time. In the procedure is necessary to assume the value $r_M \geq r_{p,s} \ \forall \ r_{p,s}$, such that $r_{p,s} = r_M$, if and only if s did not solve p.

Finally, the value $\rho_s(\tau)$, which provides a performance overview of each $s \in S$ is given by

$$\rho_s(\tau) = \frac{1}{n_p}|p \in P : r_{p,s} \leq \tau|, \quad (5)$$

where $\tau \in [1, r_M]$. In these terms, $\rho_s(\tau)$ is the portion of problems that s can solve under cost τ. The value $\rho_s(1)$ value is the portion of problems where s performed better. The algorithm s which produced the smallest value of τ, such that $\rho_s(\tau) = 1$ is considered the most robust in S.

In this work, the mean value of the objective function of solutions obtained for the problems by the algorithms in their runs was defined as the performance measure for the Performance Profile. This value gives a representation of performance for each algorithm in all its runs.

V. Results

One of the known characteristics of DE is its fast convergence and intensification trend. Figure 1 exemplifies the mean variation of the objective function in the population along the first 200 iterations for the studied algorithms.

Figure 1 illustrates that most of the algorithms, besides the differences of their population quality, presents similarities regarding the convergence rates, which can vary according to the problem. In Fig. 1, it can be observed that some algorithms promoted a more relevant impact in DE convergence. This is the case of algorithms DEsPA (certainly for stimulating exploration by inclusion of new solutions in the population in the initial iterations), SPS-L-SHADE-EIG (relatively slower convergence, use crossover based on covariance of variables) and CCLSHADE (convergence considerably faster, population evolves based on separability of variables).

Regarding the problems not illustrated in Fig. 1, for both $D = 10$ and $D = 30$, it was verified that the algorithms convergence were similar to those illustrated in Fig. 1(a). In this case, generally only 50 iterations were needed for the algorithms to reach a level of population quality that would define the final solution. This fact demonstrate that the evaluated algorithm tends to maintain the well-known convergence characteristics of DE. However, each algorithm tends to assume a different condition regarding its population convergence if compared to others. Among the studied algorithms, winners in recent editions of CEC competition, DEsPA was the algorithm that most impacted regarding the DE convergence.

One of the characteristics proposed in L-SHADE is the reduction of the population size, also present in the algorithms that followed it. Figure 2 illustrates the difference in variation of the population size along the first 200 iterations of the algorithms to solve some problems.

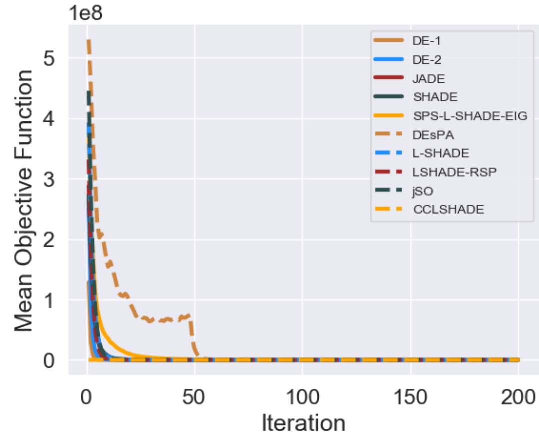
As can be observed in Fig. 2, the variation of the population size of CCLSHADE can considerably differ according to the problem. This feature is due to the grouping of non-separable variables. As the population size varies according to the number of functions evaluations and that at each iteration each variable group run the algorithm in all the population separately, it consumes evaluations. So, a large number of groups imply in a large number of evaluations at each iteration and consequently in a greater reduction in the population size. For this reason, the population reduction of CCLSHADE is faster when compared to other algorithms.

Usually, the computational cost for run EAs is defined as a maximum number of function evaluations. In this case, one of the impacts of reducing the population size, as is the case of many of the algorithms studied here, may be the run of a high number of iterations, which may cause stagnation.

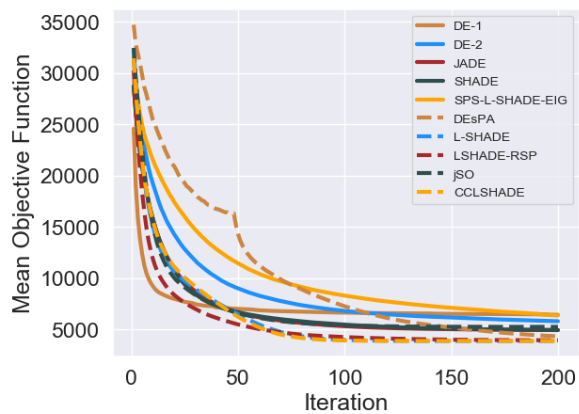
Tables I and II presents the mean of total iterations of each algorithm at each run to solve each problem under $D = 10$ and $D = 30$, respectively. Note that the algorithms that start with a relatively small population require a high number of iterations, as is the case of DEsPA and SPS-L-SHADE-EIG, even compared with those with constant and relatively small population size, as is the case of DE-1 and DE-2. In the case of DEsPA, the stagnation phase with a large population can also reduce the total of iterations if the respective parameters are properly adjusted.

The combination of the studied algorithms in D-IM was firstly analyzed by comparing the quality of the solutions produced. Figure 3 presents the Performance Profile results for $D = 10$. Note that, a zoom was applied in the results to highlight the lower values of τ such that $\rho(\tau) = 1$ and the values $\rho(\tau)$ such that $\tau = 1$.

Figure 3(a) demonstrate that for $D = 10$, DIM-1 and DIM-2 were more robust, followed by algorithms SPS-L-SHADE-EIG and DEsPA, respectively. Note that, algorithms SPS-L-SHADE-EIG and DEsPA are not the latest proposals algorithms as mentioned in Section II-A. This result demonstrates that the winners algorithms in a particular CEC competition are not necessarily better than the winners from previous



(a) $F6$



(b) $F14$

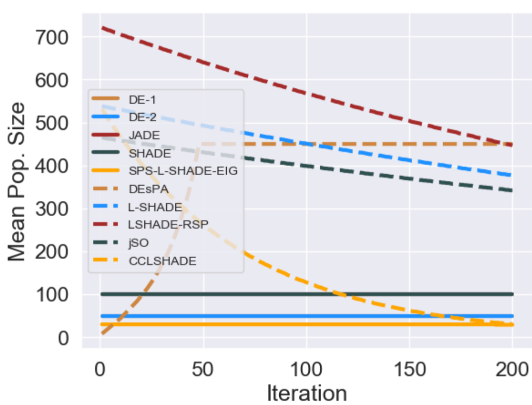
Fig. 1. Variation of mean of objective function of population along 200 iterations of each algorithm for $D = 30$.

TABLE I
Mean of maximum number of iterations of each algorithm for $D = 10$.

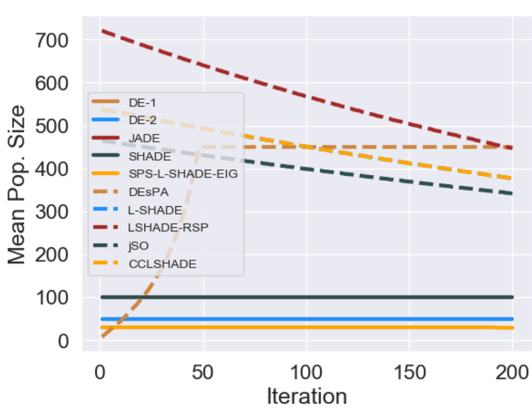
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
CCLSHADE	720	720	1272	720	720	540	558	540	1272	1272	1272	1272	1272	1272	1272
DE-1	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
DE-2	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
DEsPA	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366	8366
JADE	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333
L-SHADE	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163	2163
LSHADE-RSP	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297	1297
SHADE	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
SPS-L-SHADE-EIG	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767	7767
jSO	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145	2145

TABLE II
Mean of maximum number of iterations of each algorithm for $D = 30$.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
CCLSHADE	456	456	1564	456	456	341	331	328	1564	1564	1564	1564	1564	1564	1564
DE-1	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
DE-2	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
DEsPA	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548	23548
JADE	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
L-SHADE	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745	2745
LSHADE-RSP	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165	2165
SHADE	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
SPS-L-SHADE-EIG	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306	23306
jSO	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090	3090



(a) $F6$



(b) $F14$

Fig. 2. Variation of mean of population size over 200 iterations of each algorithm for $D = 30$.

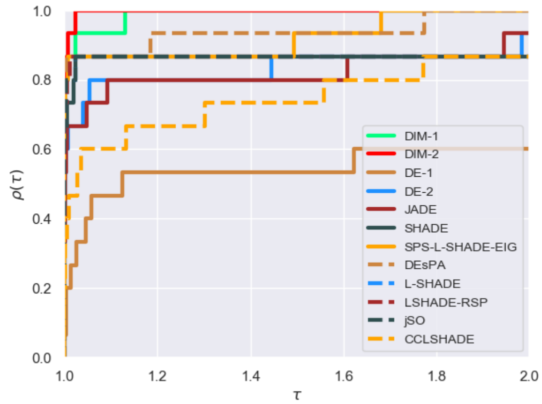
competitions, as expected. Additionally, Fig. 3(a) also suggests that DIM-1 and DIM-2 outperform the algorithms in smaller portions of problems. However, Fig. 3(b) demonstrates that L-SHADE, base for most of the studied algorithms in this work, produced the best mean of objective function for the largest number of problems in this experiment, overcoming all algorithms pointed as more robust in Fig. 3(a).

For $D = 30$, Fig. 4 presents the Performance Profile results to the algorithms comparison. Again, sub-figures in Fig. 4 highlight the lower values τ such that $\rho(\tau) = 1$ and the values $\rho(\tau)$ such that $\tau = 1$. As can be observed in Fig. 4(a), neither DIM-1 or DIM-2 is the most robust solver, but the LSHADE-RSP algorithm, followed by L-SHADE and SPS-L-SHADE-EIG. Individual algorithms also outperformed DIM-1 and DIM-2 to solve smaller portions of problems.

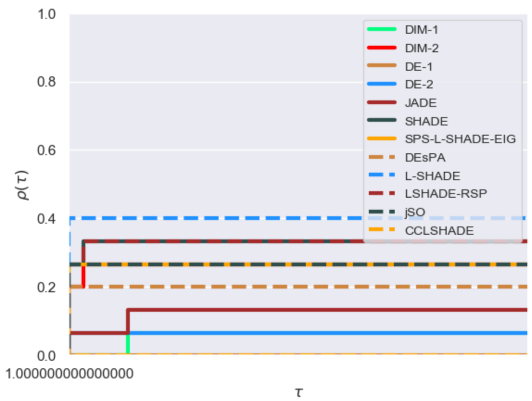
Figure 4(b) demonstrates that, as for $D = 10$, also for $D = 30$, the L-SHADE was the algorithm that produced the best mean of objetive function for the largest number of problems in this experiment.

Considering that the increase in a problem dimension implies in increasing complexity, it is interesting to access a considerable number of domain regions to solve the problem. So, DIM-1 and DIM-2 were evaluated with a larger population, in this case $POP = 400$. Figure 5 presents results of Performance Profile applied to compare separately the algorithms with DIM-1 and DIM-2 with $POP = 300$ and $POP = 400$. Figure 5 highlights the values of τ referring to DIM-1 and DIM-2 such that $\rho(\tau) = 1$.

Comparing Fig. 5(a) and 5(b), it is possible to observe that even under the increase of their populations, DIM-1 and DIM-2 were overcome by some individual algorithms in terms of

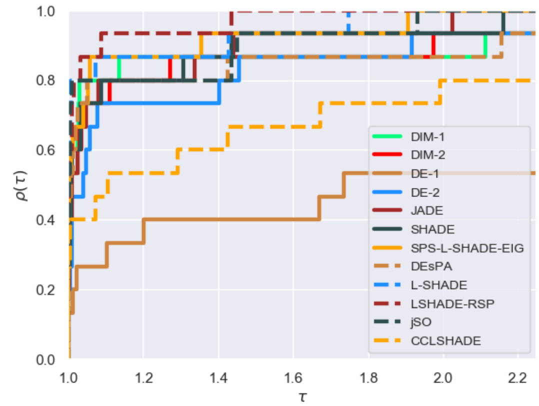


(a) Lower values of $\tau \mid \rho(\tau) = 1$.

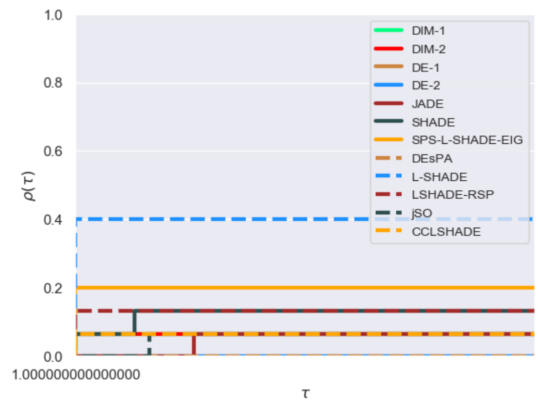


(b) Values $\rho(\tau) \mid \tau = 1$.

Fig. 3. Performance Profile for $D = 10$.



(a) Lower values of $\tau \mid \rho(\tau) = 1$.



(b) Values $\rho(\tau) \mid \tau = 1$.

Fig. 4. Performance Profile for $D = 30$ and $POP = 300$ for DIM-1 and DIM-2.

robustness. However, the comparison between those figures also demonstrates that DIM-1 and DIM-2 performed better with larger populations. The τ values associated to the models such that $\rho(\tau) = 1$, in Fig. 5(b) are considerably smaller than those in Fig 5(a).

Although the graphs were not included in this work, the models DIM-1 and DIM-2 were also evaluated under $POP = 200$. The models performance were inferior than that presented under $POP = 300$. The increase in the population size really benefited the island models performance.

Regarding the distribution of solutions between islands in DIM-1 and DIM-2, Fig. 6 and 7 present the mean variations and standard deviations (vertical lines) of the total solutions directed to each island over migrations in solving the problems under $D = 10$ and $D = 30$, respectively. The values in Fig. 6 and 7 for each migration were obtained considering all problems and runs of the models. Thus, those values demonstrate the tendency of DIM-1 and DIM-2 in the complete experiment. For $D = 30$, Fig. 7 presents values referring to the models with $POP = 400$, which presented better performance, as mentioned before.

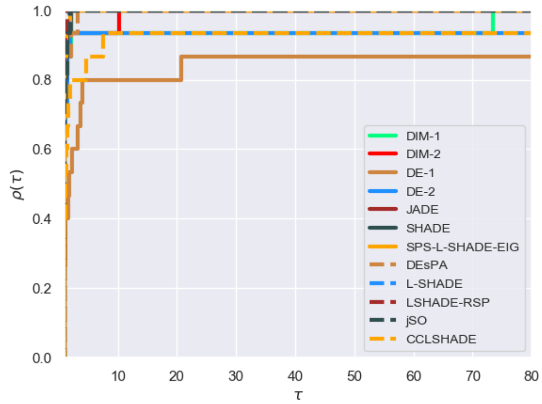
Figures 6 and 7 demonstrate that for both $D = 10$ and $D = 30$, DIM-1 and DIM-2 were able to identify differences between algorithms over their runs. In both models, it was directed a certain number of solutions for each island, which also varied in an increasing or decreasing way. Note that distinct attractiveness was defined for the islands due to their algorithms, although in the case of those that vary the population size, this operation has been disabled when they were applied in the models.

Figures 6 and 7 illustrate that, the distribution of solutions between islands in DIM-1 and DIM-2 represents the algorithms characteristics in terms of their convergences rates and quality of solutions, which is exemplified in Fig. 1.

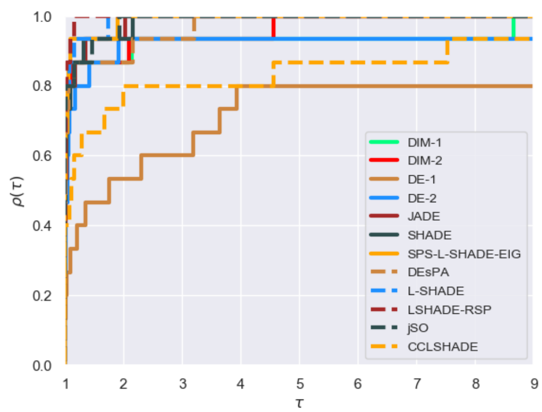
VI. Conclusion

This paper performed an evaluation between DE based algorithms, winners of competitions on bound constrained optimization problems in last editions of CEC. Also it was proposed the combination of those algorithms in two alternatives of D-IM, for hybrid implementations.

The results presented in this work demonstrated that the compared algorithms tend to maintain behavioral characteristics of DE and that new variants really can improve the performance of DE. As expected, it was also observed that a new proposed algorithms not necessarily overcome the performance of previous ones, even though both have the same

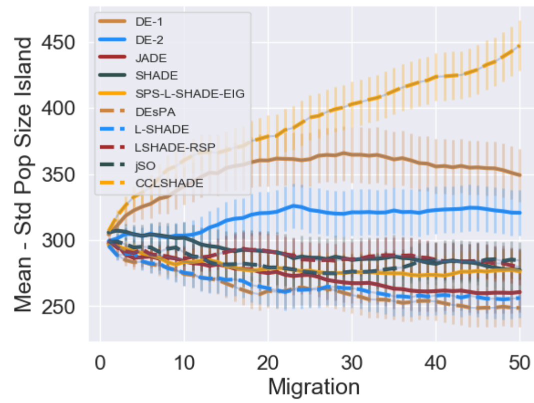


(a) $POP = 300$

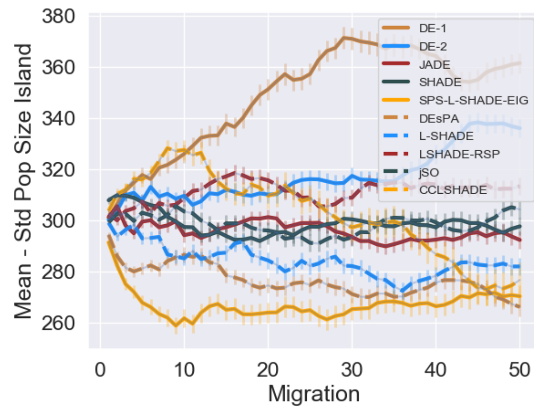


(b) $POP = 400$

Fig. 5. Values of $\tau \mid \rho(\tau) = 1$ in Performance Profile for $D = 30$ with different population size for DIM-1 and DIM-2 and individual algorithms.



(a) DIM-1



(b) DIM-2

Fig. 6. Variation of population in islands of DIM-1 and DIM-2 under $D = 10$.

algorithm as a base and they have been winners of different editions of specific competitions. This fact reinforces the need for research on structures such as DIM-1 and DIM-2 for EAs. These models can identify which algorithms may be better to solve a given problem.

It was demonstrated that the combination of DE based algorithms in DIM-1 and DIM-2 can produce better results than using just one. Also, DIM-1 and DIM-2 were satisfactory efficient to identify differences between DE based algorithms and define levels of attractiveness according to their specific strategies.

As future works, we intend to perform additional investigations on the use of similar algorithms in DIM-1 and DIM-2, initially, mainly intensifiers ones as is the case of DE. It is also intended to apply the variation in population size proposed by the algorithms when applied in DIM-1 and DIM-2 to produce new solutions over execution of the models in order to promote diversity to the population.

Acknowledgments

This research was developed with the support of the Núcleo Avançado de Computação de Alto Desempenho (NACAD) at COPPE, Federal University of Rio de Janeiro (UFRJ). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

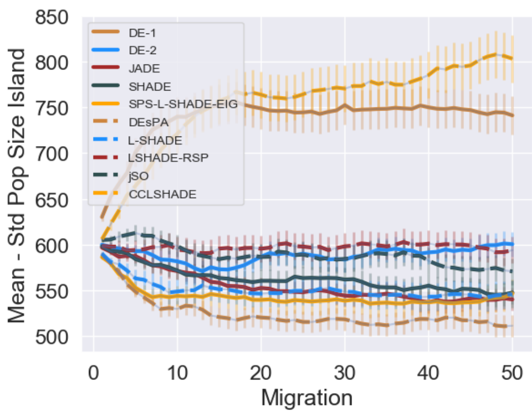
[2] F. S. Lobato, J. Valder Steffen, and A. S. Neto, “A comparative study of the application of differential evolution and simulated annealing in radiative transfer problems,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 32, pp. 518–526, December 2010.

[3] R. Mansouri, H. Torabi, M. Hoseini, and H. Morshedzadeh, “Optimization of the water distribution networks with differential evolution (de) and mixed integer linear programming (milp),” *Journal of Water Resource and Protection*, vol. 7, no. 9, pp. 715–729, 2015. [Online]. Available: www.scirp.org/journal/PaperInformation.aspx?PaperID=57833

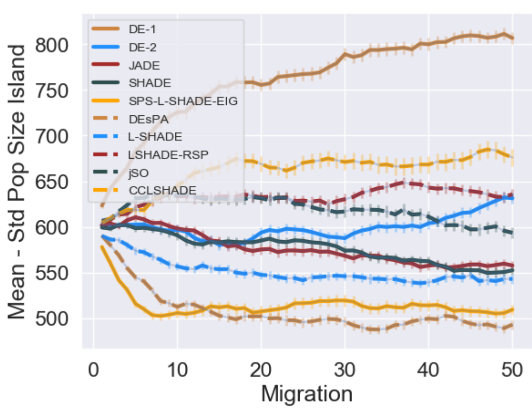
[4] D. Zaharie and F. Micota, “Revisiting the analysis of population variance in differential evolution algorithms,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1811–1818.

[5] F. Peng, K. Tang, G. Chen, and X. Yao, “Multi-start jade with knowledge transfer for numerical optimization,” in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 1889–1895.

[6] Y. Li, Z. Zhan, Y. Gong, W. Chen, J. Zhang, and Y. Li, “Differential evolution with an evolution path: A deep evolutionary algorithm,” *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1798–1810, September 2015.



(a) DIM-1



(b) DIM-2

Fig. 7. Variation of population in islands of DIM-1 and DIM-2 under $D = 30$ and $POP = 400$.

[7] J. Zhang and A. C. Sanderson, "Jade: Self-adaptive differential evolution with fast and reliable convergence performance," in *2007 IEEE Congress on Evolutionary Computation*, Sep. 2007, pp. 2251–2258.

[8] J. Zhang and A. C. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct 2009.

[9] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 397–413, April 2011.

[10] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 71–78.

[11] P. Bujok and J. Tvrdík, "Adaptive differential evolution: Shade with competing crossover strategies," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Springer International Publishing, 2015, pp. 329–339.

[12] G. Li, Q. Lin, L. Cui, Z. Du, Z. Liang, J. Chen, N. Lu, and Z. Ming, "A novel hybrid differential evolution algorithm with modified code and jade," *Applied Soft Computing*, vol. 47, pp. 577–599, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494616302903

[13] R. Tanabe and A. Fukunaga, "Evaluating the performance of shade on cec 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 1952–1959.

[14] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, July 2014, pp. 1658–1665.

[15] S.Guo, J. S. Tsai, C. Yang, and P. Hsu, "A self-optimization approach for l-shade incorporated with eigenvector-based crossover and successful-parent-selecting framework on cec 2015 benchmark set," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 1003–1010.

[16] N. Awad, M. Z. Ali, and R. G. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for cec2015 learning-based optimization," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 1098–1105.

[17] M. El-Abd, "Cooperative co-evolution using lshade with restarts for the cec15 benchmarks," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 4810–4814.

[18] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1311–1318.

[19] J. Brest, M. S. Maucec, and B. Boskovic, "il-shade: Improved l-shade algorithm for single objective real-parameter optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 1188–1195.

[20] V. Stanovov, S. Akhmedova, and E. Semenkin, "Lshade algorithm with rank-based selective pressure strategy for solving cec 2017 benchmark problems," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–8.

[21] E. Alba, *Parallel Metaheuristics: A New Class of Algorithms*. Wiley, 2005.

[22] Z. M. Skolicki, "An analysis of island models in evolutionary computation," Ph.D. dissertation, Fairfax, VA, USA, 2007.

[23] M. Ruciński, D. Izzo, and F. Biscani, "On the impact of the migration topology on the island model," *Parallel Computing*, vol. 36, no. 10–11, pp. 555–571, 2010, parallel Architectures and Bioinspired Algorithms.

[24] D. Izzo, M. Rucinski, and C. Ampatzis, "Parallel global optimisation meta-heuristics using an asynchronous island-model," in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 2301–2308.

[25] R. A. Lopes, R. C. P. Silva, F. Campelo, and F. G. Guimarães, "Dynamic selection of migration flows in island model differential evolution," in *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*. ACM, 2013, pp. 173–174.

[26] M. Märtens and D. Izzo, "The asynchronous island model and nsga-ii: Study of a new migration operator and its performance," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 1173–1180.

[27] G. Duarte, A. Lemonge, and L. Goliatt, "A dynamic migration policy to the island model," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1135–1142.

[28] G. Duarte, A. Lemonge, and L. Goliatt, "A new strategy to evaluate the attractiveness in a dynamic island model," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–8.

[29] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization," Nanyang Technological University, Tech. Rep., 2014.

[30] E. D. Dolan and J. J. More, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.