Exploiting Diversity in an Asynchronous Migration Model for Distributed Differential Evolution

Ivanoe De Falco ICAR–CNR Via P. Castellino, 111 Naples, Italy ivanoe.defalco@na.icar.cnr.it

Umberto Scafuri ICAR–CNR Via P. Castellino, 111 Naples, Italy umberto.scafuri@na.icar.cnr.it

ABSTRACT

In this paper an improved version of a general-purpose asynchronous adaptive multi-population model for distributed Differential Evolution algorithm is investigated. Specifically, in addition to an asynchronous mechanism for a multi-population recombination employed to exchange information, the distributed algorithm is endowed also with an innovative mechanism able to exploit diversity for the selection of the subpopulations involved in the asynchronous communication. Moreover the model is provided with a specific updating scheme to randomly update the control parameter values.

The asynchronous migration mechanism and the adaptive procedure allow reducing the number of control parameters to be set and tuned in the distributed model respectively.

The proposed distributed algorithm has been tested on the benchmarks of the CEC2016 real parameter single objective competition without adopting any specific mechanism opportunely tailored for solving such test problems. The results compared with the basic version of the distributed algorithm reveal an improvement in the performance in most of the considered benchmarks.

CCS CONCEPTS

• Computing methodologies → Continuous space search; Distributed algorithms;

KEYWORDS

Distributed Differential Evolution, asynchronous model, performance evaluation.

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-4939-0/17/07...\$15.00

https://doi.org/http://dx.doi.org/10.1145/3067695.3084217

Antonio Della Cioppa University of Salerno Via Ponte Don Melillo, 1 Fisciano (SA), Italy adellacioppa@unisa.it

Ernesto Tarantino ICAR–CNR Via P. Castellino, 111 Naples, Italy ernesto.tarantino@na.icar.cnr.it

ACM Reference format:

Ivanoe De Falco, Antonio Della Cioppa, Umberto Scafuri, and Ernesto Tarantino. 2017. Exploiting Diversity in an Asynchronous Migration Model for Distributed Differential Evolution. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017*, 8 pages. https://doi.org/http://dx.doi.org/10.1145/3067695.3084217

1 INTRODUCTION

During the last decade Differential Evolution (DE) has been widely investigated for solving global continuous optimization problems. The motivation behind this success within the research communities is twofold. On the one side DE success requires few control parameters and is simpler to implement with respect to most other Evolutionary Algorithms (EAs). On the other side, such a technique is characterized by low complexity [15] and outperforms, in terms of final accuracy, convergence speed, and robustness, many of the already existing stochastic and direct search global optimization techniques [6, 14, 42].

Nevertheless, its performance depends on the choice of the transformation scheme to produce the trial vector and on the setting of the control parameters to avoid premature convergence or an undesired stagnation condition [13, 26, 42], according to both experimental investigations [25] and theoretical analyses [46]. With the aim to improve the performance, the outcome of the DE investigation has given rise to a multitude of sequential approaches on how to properly choose the trial vector generation strategy and to avoid the manual tuning of these control parameter values [9, 41, 43, 46]: i) deterministic parameter control which takes place when the value of a parameter is altered by some deterministic rule [13, 22, 35, 52, 55]; ii) adaptive parameter control when a form of feedback from the search is used to determine the direction and/or the magnitude of the changes in the parameters [8, 29, 37, 49, 54, 56]; iii) self-adaptive parameter control which exploits an evolutionary process to implement the self-adaption of the parameters. This means that the parameters are encoded in the chromosome and undergo the mutation and recombination operators [7, 47, 53].

In the last years much research activity has been dedicated to the development of distributed Differential Evolution (dDE) approaches [15–20, 23, 27, 30, 33]. The motivation behind this research line can be found in the fact that they are able to explore a search space

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

more evenly, and can fight population stagnation thanks to a better capability of preserving an overall higher diversity [2].

Based on the previous considerations, we introduced a generalpurpose asynchronous adaptive multi-population algorithm for distributed Differential Evolution, referred to as Asynchronous Adaptive Multi-Population Model for dDE (AsAMP-dDE) [21], characterized by an asynchronous mechanism for the migration and for a multi-population recombination employed to exchange information. The motivation of this choice was that the use of an asynchronous migration mechanism may result advantageous with respect to its synchronous counterpart in terms of result quality [24].

Differently from AsAMP-dDE in which the migration among the subpopulations occurs on a probabilistic basis, in the algorithm proposed within this paper, hereinafter referred to as Diversitybased Adaptive Multi-Population Model for dDE (AsDivAMP-dDE), the policy adopted for the selection of the subpopulations involved in the asynchronous communication is guided by the diversity between the current global best individual and the local best ones. In particular, the subpopulations selected for such a communication are those containing the current global best and the most 'diverse' current local best. This entails that the communication is potentially possible between any pair of subpopulations rather than only among subpopulations that are neighboring on the basis of the migration topology as occurs in AsAMP-dDE.

The mechanism for exploiting the diversity is a very important factor for enhancing the performance of evolutionary algorithms [12, 28]. In fact, diversity-preserving mechanisms are able to improve the global exploration of the search space and enable crossover to find dissimilar individuals for recombination.

AsAMP-dDE borrowed some migration issues and one of the adaptive control parameter schemes adopted by AIM-dDE [19]. The adaptive procedure, based on a fitness-based measurement, aims to enhance the exploration in the parameter space in those subpopulations with lower improvements in performance. The hope is that in this way such subpopulations will find good individuals and parameter values, thus improving their performance. In general the updating scheme is employed for a dynamic setting of both the mutation and the crossover parameters, where the subpopulation size is assumed to be tuned in advance and kept fixed throughout the evolution process.

The employment of the asynchronous migration and of the adaptive scheme allows reducing the number of the control parameters to be set and tuned and this represents a noticeable strength point of the proposed distributed model.

The performance of AsDivAMP-dDE algorithm has been investigated on the CEC 2016 competition test suite on learningbased real-parameter single objective optimization problems.

In the following, Section 2 describes the classical DE technique. Some works on the asynchronous mechanism are reported in Section 3; Section 4 illustrates AsDivAMP-dDE algorithm with its adaptive scheme and the new adopted asynchronous migration mechanism. In Section 5 the experimental results obtained by this algorithm are presented and discussed. The final section reports the conclusions and the future works.

2 DIFFERENTIAL EVOLUTION

DE [40, 46] is one of the most efficient and reliable population– based stochastic techniques for solving optimization problems in continuous search domains. The population of DE made up of a fixed number of potential solutions is randomly initialized within a multidimensional search space. Then such a population is evolved over time by using three operators of mutation, crossover, and selection to explore the search space and to locate the optima of the fitness function, denoted with Φ , which evaluates the optimality of a solution.

More specifically, given a minimization problem within a Ddimensional search space of real parameters, DE deals with it by starting with a population of *l* randomly chosen potential solution vectors \mathbf{x}_i each made up by *D* real values, one for each problem dimension. The population is evolved from one generation t to the next creating new individuals by combining vectors chosen within the current population (mutation). The mutant vectors are then mixed with a predetermined target vector (crossover operator) to create the trial vector. Many different transformation schemes have been introduced by the inventors to produce the candidate trial vector [40, 46]. To explicit the strategy they established a notation for each DE technique with a string like DE/base/num/cross. In it DE stands for Differential Evolution, base is a string which denotes the vector to be perturbed (best = the best individual in the current population, *rand* = a randomly chosen one, *rand-to-best* = a random one, but the current best takes part in the perturbation too), num is the number of difference vectors taken for perturbation of base (either 1 or 2), while cross is the crossover method (exp = exponential, bin = binomial). A difference vector is a subtraction between two solution vectors within the current population. For example the DE/best/2/bin model involves that the best individual is perturbed by using two difference vectors and by applying binomial crossover. More specifically, to create the new *i*-th individual in the next population four integer numbers r_1 , r_2 , r_3 and r_4 in $[1, \ldots, l]$ differing from one another and different from *i* are randomly chosen. Once the mutant vector $\mathbf{x}_i^* = \mathbf{x}_{hest} + F \cdot ((\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) + (\mathbf{x}_{r_3} - \mathbf{x}_{r_4}))$ where F is the scale which controls the magnitude of the differential variation, is created, it will undergo binomial crossover with \mathbf{x}_i .

To this scope, another integer number *s* in the range [1, D] is randomly generated. Starting from the mutant vector a random number ρ in [0.0, 1.0] is generated, and if this is lower than the crossover rate *CR* (control parameter of the DE set by the user, in the same range as ρ) or the position *j* under account is exactly *s*, then the *j*-th gene of the trial \mathbf{x}'_{j} is generated as:

$$x_{i,j}' = x_{i,j}^* = x_{best,j} + F \cdot ((x_{r_1,j} - x_{r_2,j}) + (x_{r_3,j} - x_{r_4,j}))$$
(1)

otherwise the gene of the original vector is kept: $\mathbf{x}'_{i,j} = \mathbf{x}_{i,j}$. Finally, the selection phase happens: this trial individual \mathbf{x}'_i is compared against \mathbf{x}_i in the current population and, if fitter, replaces it in the next population, otherwise the original one survives into the new population. Such a comparison is accomplished on the basis of Φ . This basic scheme is repeated for a maximum number of generations t_{max} or until some stopping criterion is satisfied.

Exploiting Diversity in an Asynchronous Migration Model for Distributed DE GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

3 WORKS ON ASYNCHRONOUS MECHANISM

An excursus of the available literature reveals that asynchronous distributed EA models have not been extensively investigated [32]. Some authors state that "the main feature of synchronous methods are the so-called synchronization points. When a worker reaches the synchronization point it stops and waits until all workers reach the same point. This means that workers must wait for the slowest worker ... Idle time can be avoided by using the asynchronous approach where workers do not wait for each other. When a worker finishes a task, it reports the results and immediately starts a new task" [38]. Analogously other researchers consider an algorithm to be synchronous when only a processing element may access a data structure, and asynchronous when multiple processing elements may access the same data structure (individual) at the same time [32]. This means that the asynchronism is introduced just to reduce the idle times which inevitably occur when processing elements must wait for data structure to be available. These definitions should not be confused with the synchronous/asynchronous migration in distributed EAs. During the years the use and the meaning of the synchronous/asynchronous nature of the migration process in evolutionary computation have given rise to several different interpretations of the asynchronism mechanism [4, 31].

Some authors [1, 2, 48] link the synchronism to the physical communication process among the demes. In this way, the migration synchronism is strictly related to the physical time and, consequently, to the particular hardware used. For example, Alba and Troya [3] attested that "sync: is a flag indicating whether the algorithm is performing regular blocking input/output communications from/to another islands, or whether individuals are integrated in the receiving population whenever they arrive from its neighbors". It is then clear that in the case of asynchronous communication, such a way of implementing the migration could influence the algorithmic behavior, depending on the specific hardware used, as evidenced by the same authors in another paper [4]: "in general, if all the machines are of the same type ... no differences should appear in the evaluation effort or algorithmic behavior ... This is not necessarily true if processors of different types are being used". In other words, if the physical communication process among the islands is asynchronous and the processors are the same, the migration will behave in a synchronous way.

Other authors proposed asynchronous migration mechanisms related to a deliberate algorithmic choice with no reference to how the islands communicate at a hardware level. Differently from the synchronous migration that takes place after a specified number of generations of the evolutionary process, in such an asynchronous model the migration occurs when a migration criterion is fulfilled [10] and the particular hardware used has no influence. It is to point out that here the migration is asynchronous in the sense that islands communicate at different times of the evolutionary process, i.e., a migration interval cannot be defined, while the underlying physical communication mechanism can be indifferently either synchronous or asynchronous (blocking vs. non-blocking communications). As an example, some authors control the migration through a random or probabilistic process [5, 39, 50]. In other papers, on each island migration is triggered by the fitness or the diversity distribution [10, 36].

Our asynchronous model follows this last mechanism since the migration is driven by the occurrence of a specific condition during the evolution.

4 THE AsDivAMP-dDE ALGORITHM

AsDivAMP-dDE is a novel adaptive multi-population distributed model in which the population is subdivided into several semiisolated subpopulations, namely *demes*. With reference to the neighborhood topology, this distributed structure may be classified as following the *island model* (fully connected demes) [44, 51]. The information exchange among neighboring subpopulations is handled by the migration policy [11]. AsDivAMP-dDE uses an asynchronous migration mechanism, which occurs when specific diversity conditions occur. It is also endowed with an asynchronous multi-population recombination instead of the probabilistic exchange mechanism derived from AsAMP-dDE. It should be also noted that in AsDivAMP-dDE no migration topology is arranged since the communication guided by the diversity can involve any pair of subpopulations.

Execution of the algorithm begins with an initialization process for each node p which hosts an instance of a DE. Hypothesizing that there are N nodes, N subpopulations $\{P_1(t = 0), \ldots, P_p(t = 0), \ldots, P_N(t = 0)\}$, each one composed of l individuals, are randomly sampled and each of them is allocated on a different node.

The subpopulation $P_p(t)$ of each node p performs a sequential DE until t equals t_{max} generations. As a consequence, each subpopulation is updated from a generation to the next one by means of the steps typical of the DE chosen.

At each generation t > 0 each node sends/receives the individual with the best fitness to/from all the other subpopulations so to individuate the global best individual GB(t). If more than one global best exists, a random one is selected. The diversity between each local best individual $LB_p(t)$, $\forall p \in \{1, ..., N\}$ and the global best one is evaluated as the euclidean distance between the components of the involved individuals. The communication takes place between the subpopulation with GB(t) and the subpopulation entailing the most diverse (distant) local best one, represented as MDLB(t). If there is more than one MDLB(t) individual, a random one is picked up. The set of individuals of each of the two subpopulations $P_p(t)$ subject to migration, i.e., the *propagule*, indicated as $M_{P_p(t)}$, is determined by collecting the individuals of $P_p(t)$ which are fitter than its current average fitness $\langle \Phi_p(t) \rangle$:

$$\mathcal{M}_{P_p(t)} = \{x_i^p(t) \in P_p(t) \mid \\ \Phi(x_i^p(t)) > \langle \Phi_p(t) \rangle = \frac{1}{l} \sum_{j=1}^l \Phi(x_j^p(t))\} \quad (2)$$

where $\Phi(x_i^p(t))$ is the fitness associated to the individual $x_i^p(t)$ and ">" is a binary relation stating that the left member is fitter than the right member.

At this point, for both the subpopulations, a *recombination* population $\Pi_p(t)$, formed by both the native and the individuals belonging to the arrived propagule, is considered for the recombination:

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

$$\Pi_p(t) \equiv P_p(t) \cup \mathcal{M}_{P_{\tilde{p}(t)}} \tag{3}$$

The best element of the propagule $M_{P_{\tilde{p}(t)}}$ associated to the subpopulation with GB(t) replaces the worst individual of the target subpopulation $P_p(t)$ containing MBLB(t). Subsequently, for both the subpopulations, to create each of the *l* individuals of the new corresponding subpopulation $P_p(t + 1)$, the recombination is applied to $\Pi_p(t)$ by selecting the target individual in $P_p(t)$, by creating the mutant individual in $\Pi_p(t)$ and, finally, by obtaining the trial individual through their recombination. The offspring of this recombination are inserted into the new corresponding subpopulation $P_p(t + 1)$ if they are fitter than the respective target individuals.

4.1 Adaptive Procedure

We recall here the three steps of the procedure on which the adaptive model is based considering, without loss of generality, the case of minimization problems:

- (1) At the beginning of the evolution for each subpopulation random values for mutation and crossover parameters F and CR are sampled from two independent uniform distributions U(0.1, 1.0). This distribution has been employed in the following each time a parameter is randomly generated.
- (2) At each generation, and before the new recombination subpopulations Π_p(t) are created on the involved nodes, for each subpopulation the local improvement Δ(Φ_p(t)) of the average fitness is evaluated as the difference between the average fitness values after the last generation and at the current generation (for the first generation time only, the initial generation is taken into account as 'last generation'):

$$\Delta \langle \Phi_{p}(t) \rangle = \langle \Phi_{p}(t-1) \rangle - \langle \Phi_{p}(t) \rangle \tag{4}$$

(3) According to the adopted adaptive procedure, i. e. *RandAvg* scheme borrowed from AIM–dDE, the average of all the local improvements, i.e., the global average improvement (Δ(Φ_p(t))), is computed:

$$\left\langle \Delta \langle \Phi_{p}(t) \rangle \right\rangle = \frac{1}{N} \sum_{p=1}^{N} \Delta \langle \Phi_{p}(t) \rangle \tag{5}$$

In each subpopulation for which $\Delta \langle \Phi_p(t) \rangle < \langle \Delta \langle \Phi_p(t) \rangle \rangle$ the control parameter values are randomly replaced.

This strategy allows that some of the most promising parameter values can be further exploited while, at the same time, assures the exploration of new potentially propitious search parameter values. In fact, the subpopulations with the greater average fitness improvements will keep on exploiting the current parameter values, whereas those with the lower improvements will very likely explore new parameter space domains. The number of updated parameter couples varies dynamically over generations according to the fitness feedback from the search so that the updating scheme is characterized by an adaptive parameter control behavior. **Algorithm 1** Pseudo-code of AsDivAMP-dDE on a generic node p with $p \in \{1, ..., N\}$

t = 0 $\Pi_{p}(t) = \emptyset; \ \mathcal{M}_{P_{\tilde{p}}}(t) = \emptyset$ **randomly initialize** a subpopulation $P_p(t) = {\mathbf{x}_1^p(t), \dots, \mathbf{x}_l^p(t)}$ evaluate the fitness $\Phi(\mathbf{x}_i^p(t), \forall i \in \{1, \ldots, l\}$ randomly initialize the values for F and CR from two independent U(0.1, 1.0)while halting conditions are not satisfied do t = t + 1 $\Pi_{p}(t) \equiv P_{p}(t-1) \cup \mathcal{M}_{P_{\tilde{p}(t-1)}}$ **update** the subpopulation $P_{p}(t)$ using the evolutionary operators of the DE and $\Pi_p(t)$ evaluate the fitness $\Phi(\mathbf{x}_i^p(t)), \forall i \in \{1, \ldots, l\}$ send/receive the current best fitness individual to/from all the other subpopulations evaluate the diversity of all the local best individuals with the current global best individual GB(t) and find the most diverse, be it MDLB(t) $\mathcal{M}_{P_{\tilde{o}}}(t) = \emptyset$ if $P_p(t)$ contains GB(t) then **create** the propagule $\mathcal{M}_{Pp}(t)$ send/receive the propagule $\mathcal{M}_{P_p}(t) / \mathcal{M}_{P_p}(t)$ to/from the island that contains MDLB(t)end if if $P_{p}(t)$ contains MDLB(t) then **replace** the worst element of $P_{p}(t)$ with GB(t)**create** the propagule $\mathcal{M}_{P_p}(t)$ receive/send the propagule $\mathcal{M}_{P_{\tilde{p}}}(t) / \mathcal{M}_{P_{p}}(t)$ from/to the island that contains GB(t)end if end while

The pseudo-code of this model for a generic node is outlined in Algorithm 1.

5 EXPERIMENTS

The distributed algorithm AsDivAMP-dDE is written in C language, and Message Passing Interface (MPI) [45] is used for the communications. Throughout the experiments the DE/*best*/2/*bin* [40] mechanism has been used. All the experiments have been carried out on an Intel Core i7 quad-core iMAC with a total memory of 16 GB. Each core has a frequency of 3.4 GHz.

As concerns the number of subpopulations and the population size, since we wish to compare AsDivAMP–dDE against AsAMP– dDE, we have decided to use the same values emplyed for this latter in [21]. This is because we wish to contrast the new migration and exchange mechanism based on diversity against the one contained in AsAMP–dDE, and, to do this in a fair way, we have to keep everything else fixed. Consequently, we have used N = 16subpopulations, and a population size equal to 128, which means to each subpopulation having a number of l = 8 individuals. Before the execution, the 16 subpopulations are distributed by means of a round-robin procedure on the four cores of the system employed for the experimentation, so that we have four running instances of AsDivAMP-dDE on each core.

The control parameters F and CR undergo the adaptive procedure described in the previous section and thus they do

not need any tuning phase. As it can be seen, the number of the parameters to set is very low, and this represents a strong point of the model.

The test problems of the CEC2016 real parameter single objective functions have been considered to evaluate the performance of the algorithm. The details of these problems, listed in Table 1, can be found in [34].

Table	1:	Test	Problems
-------	----	------	----------

Туре	No.	Description	Optimum
Unimodal function		Rotated High Conditioned Elliptic Function	100
Uniniodal function	F2	Rotated Bent Cigar Function	200
	F3	Shifted and Rotated Ackley's Function	300
Simple multimodal function	F4	Shifted and Rotated Rastrigin's Function	400
	F5	Shifted and Rotated Schwefel's Function	500
	F6	Hybrid Function 1 (N=3)	600
Hybrid function	F7	Hybrid Function 2 (N=4)	700
	F8	Hybrid Function 3 (N=5)	800
	F9	Composition Function 1 (N=3)	900
	F10	Composition Function 2 (N=3)	1,000
	F11	Composition Function 3 (N=5)	1,100
Composition function	F12	Composition Function 4 (N=5)	1,200
	F13	Composition Function 5 (N=5)	1,300
	F14	Composition Function 6 (N=7)	1,400
	F15	Composition Function 7 (N=10)	1,500

Tests on convergence performance and quality of final solution provided by AsDivAMP-dDE have been performed under the following conditions:

- problem dimension *D* = 10, 30, 50, 100;
- search range $[-100, 100]^D$;
- maximum number of function evaluations: $10,000 \cdot D$;
- 51 runs for each test problem;
- when the difference between the best solution found and the optimal was lower than or equal to 10⁻⁸, the error was treated as 0;
- the optimization is stopped upon completion of the maximum number of function evaluations.

5.1 Experimental Findings

The Tables 2 to 5 summarize the results of AsDivAMP-dDE for the different problem dimensions in terms of the mean of the best final fitness values found in the 51 runs, and of the related standard deviation. The best final fitness value found in the 51 runs is also reported. All the results are not shown here in absolute values, rather in terms of the distance of the found values from the optimum values contained in the fourth column of Table 1.

Moreover, aiming at comparing this new algorithm against the previous one used to face the same problems, i.e., the AsAMP–dDE algorithm, these tables report the values obtained by this latter algorithm too and contained in [21]. Whenever one of the two algorithms obtains better mean value than the other, the corresponding value is marked in bold in the tables.

Several comments can be made to the numerical results shown in the tables:

• As a general comment, the mean values obtained by AsDivAMP-dDE are better than those achieved by AsAMP-dDE in 43 out of 60 cases, whereas the opposite is true in 5 cases, the remaining 12 being characterized by equal final

values. Consequently, AsDivAMP-dDE actually improves its ancestor AsAMP-dDE in terms of higher-quality results.

- Unimodal functions: as far as F2 is concerned, for all the problem sizes AsDivAMP-dDE is capable of finding zero error values (i.e., smaller than 1E-08) in all the runs, and so is AsAMP-dDE. For F1, instead, for both algorithms difficulties arise as long as the problem size increases, and for D=10 only all the runs reach the optimal value. For the larger sizes, nonetheless, most runs achieve good results, and just few unlucky runs cause an increase in the mean values. This is confirmed by the median values, not reported here, that are much lower than the corresponding mean ones.
- Simple multimodal functions: AsDivAMP-dDE and AsAMPdDE meet increasing difficulty when passing from F3 to F4 and from this latter to F5. Optimal values are never reached at any problem size. The standard deviations for F5 are very high, meaning that some very bad runs are present. This is confirmed by the worst values, not reported here. On F3 both algorithms reach about the same values for all the problem sizes.
- Hybrid functions: for this set of test functions too AsDIVAMP-dDE faces difficulties, and in no run the global best is reached, even at the smallest dimension. F7 seems to be much easier to solve than the other functions for our algorithm. The same considerations hold true for AsAMPdDE.
- Composition functions: for this function set too AsDIVAMPdDE is never able to reach the global best at all problem dimensions, as it is the case for AsAMP-dDE. F14 seems to be the hardest function for both algorithms to tackle. It is worth noting that for F15 the local minimum of 100 is reached in almost all the runs at all the problem dimensions. On F9 and F12 the behavior of both algorithms seems to be quite independent of the problem size, all the statistics being comparable at the different dimensions.

5.2 Statistical Analysis

From the tables 2, 3, 4, and 5 it can be seen that most of the times the mean values obtained by AsDivAMP-dDE are better than those of the original AsAMP-dDE.

To investigate whether the former is superior from a statistical point of view or there is a statistical equivalence between them, a statistical analysis has been carried out.

Our zero hypothesis is that there is a statistical equivalence between the two algorithms. To assess whether or not this holds true, firstly a test about normality of the distributions of the variables has been effected. The result has been that the distributions are not normal, which leads us to use a nonparametric test. Given than we have just two algorithms here, no tests as Friedman, Aligned Fredman, or Quade can be used. Rather, we have availed ourselves of the Wilcoxon signed ranked test with $\alpha = 0.05$.

The statistic for Wilcoxon p-value has turned out to be equal to 1557.5 with 1 and 58 degrees of freedom. The resulting p-value is 0.0000. This last value means that the hypothesis of statistical

Table 2: Statistical Results of the 10-D Benchmark Functions, Averaged over 51 Independent Runs.

AsAMP-dDE			AsDivAMP-dDE			
Test	Mean	StdDev	Best	Mean	StdDev	Best
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	2.00E+01	5.60E-02	2.00E+01	2.00E+01	1.95E-03	2.00E+01
F4	7.65E+00	3.71E+00	2.99E+00	2.77E+00	1.37E+00	1.99E+00
F5	2.76E+02	1.78E+02	6.96E+00	2.24E+02	1.72E+02	3.66E+00
F6	5.54E+01	6.26E+01	1.20E+00	1.99E+01	5.03E+01	0.00E+00
F7	6.75E-01	4.53E-01	1.90E-02	8.68E-02	9.64E-02	0.00E+00
F8	2.17E+01	4.60E+01	1.00E-03	2.84E+00	6.65E+00	1.02E-02
F9	1.00E+02	8.10E-02	1.00E+02	1.00E+02	6.07E-02	1.00E+02
F10	2.20E+02	6.48E+00	2.17E+02	2.19E+02	7.33E+00	2.17E+02
F11	1.31E+01	4.10E+01	3.70E+00	8.76E+00	4.16E+01	1.18E+00
F12	1.02E+02	9.19E-01	1.01E+02	1.02E+02	5.49E-01	1.01E+02
F13	3.09E+01	3.41E+00	2.57E+01	2.91E+01	3.76E+00	2.22E+01
F14	2.13E+03	1.50E+03	1.00E+02	4.95E+02	1.00E+03	1.00E+02
F15	1.00E+02	0.00E+00	1.00E+02	1.00E+02	9.85E-15	1.00E+02

Table 3: Statistical Results of the 30-D Benchmark Functions, Averaged over 51 Independent Runs.

AsAMP-dDE				AsDivAMP-dDE		
Test	Mean	StdDev	Best	Mean	StdDev	Best
F1	7.52E+01	5.10E+02	1.00E-04	1.67E+01	1.15E+02	4.80E-08
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	2.04E+01	2.02E-01	2.00E+01	2.00E+01	7.97E-02	2.00E+01
F4	6.09E+01	2.02E+01	3.08E+01	3.05E+01	1.21E+01	2.09E+01
F5	3.29E+03	6.68E+02	1.91E+03	2.63E+03	6.87E+02	1.57E+03
F6	1.22E+03	3.86E+02	5.51E+02	9.26E+02	5.58E+02	3.97E+02
F7	5.05E+00	7.83E-01	3.70E+00	3.75E+00	4.69E-01	3.06E+00
F8	5.08E+02	2.75E+02	1.53E+01	3.30E+02	1.82E+02	1.50E+02
F9	1.04E+02	3.72E-01	1.03E+02	1.03E+02	2.12E-01	1.03E+02
F10	9.50E+02	3.42E+02	2.18E+02	6.54E+02	1.73E+02	4.22E+02
F11	3.95E+02	1.78E+02	3.01E+02	3.01E+02	1.93E-02	3.01E+02
F12	1.08E+02	1.00E+00	1.06E+02	1.06E+02	8.75E-01	1.04E+02
F13	1.19E+02	5.40E+00	1.03E+02	1.10E+02	7.22E+00	9.65E+01
F14	3.22E+04	1.05E+03	3.11E+04	3.18E+04	8.03E+02	3.11E+04
F15	1.00E+02	0.00E+00	1.00E+02	1.00E+02	0.00E+00	1.00E+02

Table 4: Statistical Results of the 50-D Benchmark Functions, Averaged over 51 Independent Runs

AsAMP-dDE			AsDivAMP-dDE			
Test	Mean	StdDev	Best	Mean	StdDev	Best
F1	3.96E+03	1.98E+04	3.81E-01	3.88E+03	1.95E+04	3.65E-06
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	2.06E+01	1.41E-01	2.03E+01	2.02E+01	2.59E-01	2.00E+01
F4	1.44E+02	3.02E+01	9.45E+01	1.03E+02	2.12E+01	7.76E+01
F5	7.45E+03	1.75E+03	3.66E+03	5.47E+03	9.85E+02	3.73E+03
F6	2.63E+03	1.19E+03	1.50E+03	2.52E+03	8.74E+02	1.52E+03
F7	1.49E+01	9.72E+00	9.09E+00	8.22E+00	1.22E+00	6.85E+00
F8	1.42E+03	1.12E+03	4.78E+02	1.22E+03	6.79E+02	6.86E+02
F9	1.07E+02	5.98E-01	1.06E+02	1.05E+02	4.93E-01	1.04E+02
F10	2.03E+03	3.37E+02	1.36E+03	1.49E+03	3.84E+02	9.87E+02
F11	3.00E+02	2.00E-04	3.00E+02	3.01E+02	1.65E-01	3.00E+02
F12	1.12E+02	1.23E+00	1.10E+02	1.09E+02	9.86E-01	1.08E+02
F13	2.12E+02	9.37E+00	1.90E+02	2.04E+02	7.13E+00	1.85E+02
F14	6.74E+04	4.17E+03	5.92E+04	5.34E+04	9.09E+03	4.95E+04
F15	1.00E+02	0.00E+00	1.00E+02	1.00E+02	2.01E-15	1.00E+02

Exploiting Diversity in an Asynchronous Migration Model for Distributed DE GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

AsAMP-dDE			AsDivAMP-dDE			
Test	Mean	StdDev	Best	Mean	StdDev	Best
F1	3.17E+04	1.50E+05	1.43E+01	9.67E+03	4.21E+04	8.32E-04
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	2.08E+01	2.24E-01	2.00E+01	2.04E+01	3.63E-01	2.00E+01
F4	4.07E+02	5.95E+01	2.79E+02	3.24E+02	3.89E+01	2.55E+02
F5	1.74E+04	3.16E+03	1.16E+04	1.35E+04	1.83E+03	1.09E+04
F6	8.72E+03	1.09E+04	4.27E+03	8.95E+03	1.91E+04	4.77E+03
F7	3.08E+01	4.21E+00	2.61E+01	2.56E+01	1.89E+00	2.28E+01
F8	3.38E+03	3.19E+03	1.46E+03	4.74E+03	6.77E+03	3.05E+03
F9	1.13E+02	1.15E+00	1.11E+02	1.11E+02	1.17E+00	1.09E+02
F10	4.50E+03	5.26E+02	3.69E+03	4.48E+03	4.78E+02	3.68E+03
F11	1.59E+03	4.60E+02	1.22E+03	1.79E+03	1.76E+02	1.67E+03
F12	1.21E+02	1.60E+00	1.17E+02	1.23E+02	1.36E+00	1.20E+02
F13	4.47E+02	1.03E+01	4.27E+02	4.21E+02	1.19E+01	4.01E+02
F14	1.09E+05	1.43E+01	1.09E+05	1.09E+05	9.69E+00	1.09E+05
F15	1.01E+02	3.66E+00	1.00E+02	1.00E+02	9.23E-01	1.00E+02

Table 5: Statistical Results of the 100-D Benchmark Functions, Averaged over 51 Independent Runs

equivalence between the two algorithms can be statistically rejected, hence AsDivAMP-dDE is statistically superior to AsAMP-dDE.

6 CONCLUSIONS AND FUTURE WORKS

It has been recognized in literature that during the evolutionary process the employment of distributed variants of the DE algorithm allow improving the performance. To this aim, in this paper we have proposed an asynchronous adaptive multi–population model for dDE, i.e. AsDivAMP–dDE, endowed with a novel mechanism for exploiting the diversity for the selection of the subpopulations involved in the migration. The adaptive process is tied to the generation, is guided by a performance measure that relies on the average fitness improvements of each subpopulation between two successive generations.

It is to point out that the strength points of AsDivAMP-dDE rely on its general-purpose nature and, by adopting an asynchronous migration and an adaptive scheme, on the request of a low number of control parameters to be set and tuned.

Such an adaptive algorithm has been evaluated on learningbased single objective optimization functions derived from the CEC 2016 competition with good performance with all the investigated benchmarks when compared with the basic version of the asynchronous model, i.e., AsAMP-dDE. It is worth noting that the proposed algorithm is a simple asynchronous adaptive distributed algorithm that does not use any specific mechanism opportunely tailored for the examined tests.

Future works will aim to explore other adaptive mechanisms and different transformation schemes of DE. Furthermore, a more complete comparison with other state–of–the art algorithms to further ascertain the effectiveness of the proposed adaptive model will be performed. Finally, efforts will be devoted to evaluate the capability of this algorithm in solving a dedicated group of problems, rather than a broad variety of benchmarks representing different kinds of difficulties.

REFERENCES

- E. Alba, G. Luque, and S. Nesmachnow. 2013. Parallel metaheurisitcs: recent advances and trends. Int. Trans. Oper. Res. 20, 1 (Jan. 2013), 1–48.
- [2] E. Alba and M. Tomassini. 2002. Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation 6 (2002), 443–462.
- [3] E. Alba and J. M. Troya. 2000. Influence of the migration policy in parallel distributed GAs with structured and panmictic populations. *Appl. Intell.* 12, 3 (2000), 163–181.
- [4] Enrique Alba and José M. Troya. 2001. Analyzing synchronous and asynchronous parallel distributed genetic algorithms. *Future Generation Computer Systems* 17, 4 (Jan. 2001), 451–465.
- [5] J Apolloni, J García-Nieto, E Alba, and G Leguizamón. 2014. Empirical evaluation of distributed Differential Evolution on standard benchmarks. *Appl. Math. Comput.* 236 (2014), 351–366.
- [6] A. Basak, S. Das, and K. C. Tan. 2013. Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance based selection. *IEEE Transactions on Evolutionary Computation* 17, 5 (2013), 666–685.
- [7] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. 2006. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation* 10, 6 (2006), 646–657.
- [8] J. Brest and M. S. Maučec. 2011. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Computing* 15, 11 (2011), 2157–2174.
- [9] J Brest, A Zamuda, B. Bošković, and V Žumer. 2008. An analysis of the control parameters' adaptation in DE. In *Studies in Computational Intelligence*, U. K. Chakraborty (Ed.). Vol. 143. Springer, 89–110.
- [10] P. Bujok. 2013. Synchronous and asynchronous migration in adaptive differential evolution algorithms. *Neural Netw. World* 23, 1 (2013), 17–30.
- [11] E. Cantú-Paz. 2000. Efficient and accurate parallel genetic algorithms. Vol. 1. Kluwer Academic Publisher, Norwell, MA, USA, 81–120.
- [12] G. Chen, C. P. Low, and Z. Yang. 2009. Preserving and exploring genetic diversity in evolutionary programming algorithms. *IEEE Transactions on Evolutionary Computation* 13, 3 (2009), 661–673.
- [13] S. Das, A. Konar, and U. K. Chakraborty. 2005. Two improved differential evolution schemes for faster global search. Proceedings of Genetic and Evolutionary Computation Conference 5 (2005), 991–998.
- [14] S. Das, A. Konar, U. K. Chakraborty, and A. Abraham. 2009. Differential evolution with a neighborhood-based mutation operator: a comparative study. *IEEE Transactions on Evolutionary Computation* 13, 3 (2009), 526–553.
- [15] S. Das and P. N. Suganthan. 2011. Differential evolution: A survey of the stateof-the-art. IEEE Transactions on Evolutionary Computation 15, 1 (2011), 4–31.
- [16] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino. 2007. Distributed differential evolution for the registration of remotely sensed images. In Proceedings of the 15th IEEE Euromicro Conference on Parallel, Distributed and Network-based Processing. IEEE Press, 358–362.
- [17] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino. 2007. Satellite image registration by distributed differential evolution. In *Lectures Notes in Computer Science*, Vol. 4448. Springer, 251–260.

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

- [18] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino. 2012. Biological invasion-inspired migration in distributed evolutionary algorithms. *Information Sciences* 207 (2012), 50–65.
- [19] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino. 2014. An adaptive invasion–based model for distributed differential evolution. *Information Sciences* 278 (2014), 653–672.
- [20] I. De Falco, A. Della Cioppa, U. Scafuri, and E. Tarantino. 2007. A Distributed Differential evolution approach for mapping in a grid environment. In Proceedings of the 15th IEEE Euromicro Conference on Parallel, Distributed and Network-based Processing. IEEE Press, 442–449.
- [21] I. De Falco, A. Della Cioppa, U. Scafuri, and E. Tarantino. 2016. An adaptive invasion-based model for distributed differential evolution. In *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE Press, Vancouver, Canada, 5010–5017.
- [22] I. De Falco, A. Della Cioppa, and E. Tarantino. 2006. Automatic classification of handsegmented image parts with Differential Evolution. In *Lecture Notes in Computer Science*. Vol. 3907. Springer, 403–414.
- [23] I. De Falco, U. Scafuri, and E. Tarantino. 2011. A distributed evolutionary approach for multisite mapping on grids. *Concurrency and Computation: Practice and Experience* 23 (2011), 1146–1168.
- [24] F F de Vega, G G Gil, and J A G Pulido. 2002. Comparing synchronous and asynchronous parallel and distributed genetic programming models. In Proceedings of the 5th European Conference on Genetic Programming. Springer-Verlag, 326–336.
- [25] A. E. Eiben, R. Hinterding, and Z. Michalewicz. 1999. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 3, 2 (1999), 124–141.
- [26] A. E. Eiben and J. E. Smith. 2003. Introduction to Evolutionary Computing. Springer Verlag, Chapter 8, 129–152.
- [27] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. 2011. Enhancing differential evolution utilizing proximity-based mutation operators. *IEEE Transactions on Evolutionary Computation* 15, 1 (2011), 99–119.
- [28] Tobias Friedrich, Pietro S. Oliveto, Dirk Sudholt, and Carsten Witt. 2009. Analysis of Diversity-Preserving Mechanisms for Global Exploration. *Evolutionary Computation* 17, 4 (2009), 455–476.
- [29] A. Ghosh, S. Das, A. Chowdhury, and R. Giri. 2011. An improved differential evolution algorithm with fitness-based adaptation of the control parameters. *Information Sciences* 181, 18 (2011), 3749–3765.
- [30] T Ishimizu and K Tagawa. 2010. A Structured Differential Evolution for Various Network Topologies. International Journal of Computers and Communications 4, 1 (2010), 2–8.
- [31] Dario Izzo, Marek Ruciński, and Christos Ampatzis. 2009. Parallel global optimisation meta-heuristics using an asynchronous island-model. In Proc. IEEE Congr. Evol. Comput. IEEE Press, 2301–2308.
- [32] D. Jakobović, M. Golub, and M. Čupić. 2014. Asynchronous and implicitly parallel evolutionary computation models. *Soft Computing* 18 (2014), 1225–1236.
- [33] G. Jeyakumar and C. S. Velayutham. 2013. Distributed mixed variant differential evolution algorithms for unconstrained global optimization. *Information Sciences* 5, 4 (2013), 275–293.
- [34] J. J. Liang, B. Y. Qu, and P. N. Suganthan. 2015. Problem definition and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Technical Report. Zhengzhou University and Nanyang Technologicl University.
- [35] C. Lin, A. Qing, and Q Feng. 2011. A comparative study of crossover in differential evolution. Swarm and Evolutionary Computation 17, 6 (2011), 675–703.
- [36] M. Munetomo, Y. Takai, and Y. Sato. 1993. An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms. In Proceedings of the 5th International Conference on Genetic Algorithms. Morgan Morgan Kaufmann, 649.
- [37] N. Noman and H. Iba. 2008. Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation* 12, 1 (2008), 107– 125.
- [38] J Olenšek, T Tuma, J Puhan, and Á Bűrmen. 2011. A new asynchronous parallel global optimization method based on simulated annealing and differential evolution. *Applied Soft Computing* 11 (2011), 1481–1489.
- [39] D. R. Penas, J. R. Banga, P. Gonzales, and R. Doallo. 2015. Enhanced parallel Differential Evolution algorithm for problems in computational systems biology. *Applied Soft Computing* 13 (2015), 86–99.
- [40] Kenneth Price and Rainer Storn. 1997. Differential evolution. Dr. Dobb's Journal 22, 4 (1997), 18-24.
- [41] K. V. Price. 1999. An introduction to differential evolution. In New ideas in optimization. McGraw-Hill Ltd., UK, 79–108.
- [42] K. V. Price, R. M. Storn, and J. A. Lampinen. 2005. Differential Evolution: a Practical Approach to Global Optimization. Springer Verlag.
- [43] J. Rönkönnen and J. Lampinen. 2005. Real-parameter optimization with differential evolution. In Proceedings of the IEEE International Conference on

Evolutionary Computation. 506-513.

- [44] K. Skolicki and K. De Jong. 2005. The influence of migration sizes and intervals on island models. In Proceedings of the Conference of Genetic and Evolutionary Computation. Association for Computing Machinery, Inc. (ACM), 1295–1302.
- [45] M. Snir, S. Otto S, Huss-Lederman, D. Walker, and J. Dongarra. 1998. MPI: The Complete Reference – The MPI Core. Vol. 1. MIT Press, Cambridge, MA, USA.
- [46] Rainer Storn and Kenneth Price. 1997. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 4 (1997), 341–359.
- [47] J. Teo. 2006. Exploring dynamic self-adaptive populations in differential evolution. Soft Computing – A Fusion of Foundations, Methodologies and Applications 10, 8 (2006), 673–686.
- [48] M. Tomassini. 2005. Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time. Springer–Verlag.
- [49] Josef Tvrdík. 2009. Adaptation in differential evolution: A numerical comparison. Applied Soft Computing 9, 3 (2009), 1149–1155.
- [50] M. Weber, F. Neri, and V. Tirronen. 2011. A study on scale factor in distributed differential evolution. *Information Sciences* 181, 12 (2011), 2488–2511.
- [51] D. Whitley, S. Rana, and R. B. Heckendorn. 1999. The island model genetic algorithm: on separability, population size and convergence. *Journal of Computing and Information Technology* 7 (1999), 33-48.
- [52] D. Zaharie. 2009. Influence of crossover on the behavior of differential evolution algorithms. Applied Soft Computing 9, 3 (2009), 62–67.
- [53] A Zamuda, J Brest, B. Bošković, and V Žumer. 2008. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution. In Proceedings of the IEEE World Congress on Computational Intelligence. 3719-3726.
- [54] J. Zhang and A. Sanderson. 2009. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 945–958.
- [55] Z. Zhao and Ponnuthurai N Suganthan. 2013. Empirical investigations into the exponential crossover of differential evolutions. *Swarm and Evolutionary Computation* 9 (2013), 27–36.
- [56] Wu Zhu, Yang Tang, Jianan Fang, and Wenbing Zhang. 2013. Adaptive population tuning scheme for differential evolution. *Information Sciences* 223 (2013), 164– 191.