Information Sciences 268 (2014) 231–254

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Distributed Estimation of Distribution Algorithms for continuous optimization: How does the exchanged information influence their behavior?

Santiago Muelas [a,*], Alexander Mendiburu [b], Antonio LaTorre [a,c], José-María Peña [a]

[a] *Dept. of Computer Systems Architecture and Technology, Universidad Politécnica de Madrid, Spain*
[b] *Intelligent Systems Group, Dept. of Computer Architecture and Technology, University of the Basque Country, Spain*
[c] *Instituto Cajal, Consejo Superior de Investigaciones Científicas, Madrid, Spain*

A R T I C L E I N F O

Article history:
Available online 24 October 2013

Keywords:
Evolutionary computation
Graphical model
Estimation of Distribution Algorithm
Island model
Continuous optimization

A B S T R A C T

One of the most promising areas in which probabilistic graphical models have shown an incipient activity is the field of heuristic optimization and, in particular, in Estimation of Distribution Algorithms. Due to their inherent parallelism, different research lines have been studied trying to improve Estimation of Distribution Algorithms from the point of view of execution time and/or accuracy. Among these proposals, we focus on the so-called distributed or island-based models. This approach defines several islands (algorithms instances) running independently and exchanging information with a given frequency. The information sent by the islands can be either a set of individuals or a probabilistic model. This paper presents a comparative study for a distributed univariate Estimation of Distribution Algorithm and a multivariate version, paying special attention to the comparison of two alternative methods for exchanging information, over a wide set of parameters and problems – the standard benchmark developed for the IEEE Workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems* of the ISDA 2009 Conference. Several analyses from different points of view have been conducted to analyze both the influence of the parameters and the relationships between them including a characterization of the configurations according to their behavior on the proposed benchmark.

1. Introduction

Estimation of Distribution Algorithms (EDAs) are a set of techniques that belong to the field of Evolutionary Computation. Since they were introduced in the 90s [32,43], the research community has put a lot of effort in their development, providing powerful algorithms which have been successfully applied to both artificial and real-world problems. In general terms, EDAs are similar to Genetic Algorithms (GAs), but their main characteristic is the use of probabilistic models to extract information from the most promising individuals of the current population (instead of using crossover or mutation operators) in order to create a new and presumably better population. The complexity of the different EDAs approaches is usually related to the probabilistic model used, and the ability of that model to identify and represent the (in) dependencies among the variables.

[*] Corresponding author.
E-mail addresses: smuelas@fi.upm.es (S. Muelas), alexander.mendiburu@ehu.es (A. Mendiburu), atorre@fi.upm.es (A. LaTorre), jmpena@fi.upm.es (J.-M. Peña).

0020-0255/$ - see front matter © 2014 Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.ins.2013.10.026

Detailed information about the main characteristics of EDAs, as well as the different algorithms that belong to this family can be found in [25,27,35,36].

The main drawback of the most complex EDAs –those that try to consider all the possible (in) dependencies among the variables– is their high computational cost. Due to this, and thanks to the modularity of EDAs, several parallel approaches have been proposed. These proposals can be divided into two groups:

- Direct parallelization or parallel EDAs (pEDAs): Those whose behavior is exactly the same of the corresponding sequential version. Their main goal is the reduction of the execution time by the parallelization of the computation of the fitness function or the construction of the graphical models [27,29,38].
- Island-based approach or distributed EDAs (dEDAs): Those that create different subpopulations and exchange information among them, trying to improve the quality of the solutions of the sequential algorithm.

In this work, we pay attention to the second approach. In this scheme, an EDA instance is executed in each island, and some information is exchanged among the islands during the execution. This information can be made up of individuals (as done in other EAs), or probabilistic models (following the rationale that EDAs use them to extract and gather information about the population). Migration of individuals is a classic approach and has proven to obtain successful results in these and other Evolutionary Algorithms [2,4,9,28]. In addition, migration of models was explicitly developed for the distributed Estimation of Distribution Algorithms [1,11,12,19,20].

Until now, most of the previous work in dEDAs has been conducted in the discrete domain, and little research has been done in comparing both migration methods (individuals versus models). In particular, in continuous optimization, as far as the authors are aware, only two studies have been carried out [10,11]. Although these papers concluded that the migration of models obtains significantly better results than the migration of individuals, the experimental scenario was restricted to (a) a limited number of problems with small dimensions and (b) a small number of parameters. In this paper, a thorough study has been conducted to analyze the behavior of the distributed approaches in the context of EDAs. This study has demonstrated that the statement of the previous studies is not correct, at least not in the field of continuous optimization, using, for this task, a standard benchmark, formal methods for conducting the analysis of the influence of the parameters, and validating the results with formal statistical procedures. Briefly, the precise objectives pursued in this work are the following:

- Conduct an extensive study of different distributed EDAs configurations over several functions and dimensions.
- Identify the parameters that have a greater influence on the final behavior and the relationships between them.
- Analyze the relationship of the parameter values of both the best and the worst configurations.
- Compare the performance of the distributed configurations against their equivalent sequential configurations.
- Compare the performance of both methods for exchanging information: individuals vs models.
- Characterize the distributed configurations according to their behavior on the proposed benchmark.

The rest of the paper is organized as follows: Section 2 presents an overview of the previous studies on EDAs and dEDAs. Section 3 describes the proposed experimental scenario. Section 4 presents and comments the results obtained and lists the most relevant facts extracted from this analysis. Finally, Section 5 contains the concluding remarks derived from this study.

2. Preliminaries

In this section the main characteristics of EDAs and dEDAs are briefly reviewed.

2.1. Estimation of Distribution Algorithms: EDAs

EDAs are stochastic heuristic search strategies that are part of the Evolutionary Computation paradigm. In EDAs, multiple solutions or individuals are created at every generation, evolving successively until a satisfactory solution is achieved. In brief, the characteristic that clearly differentiates EDAs from other evolutionary search strategies, such as GAs, is that the evolution from one generation to the next is achieved by estimating the probability distribution of a set of individuals, sampling later the induced model. This avoids the use of crossing or mutation operators, and the number of parameters required by EDAs is considerably reduced. Based on the probabilistic model considered, three main groups of EDAs can be distinguished: univariate models, which assume that variables are marginally independent; bivariate models, which accept dependencies between pairs of variables; and multivariate models, in which there is no assumption about independences.

The univariate model is the simplest model, in which independence among variables is assumed. Therefore, the joint probability distribution is defined as the product of the marginal probability of each variable. The main advantage of this model is its low computational cost, although the assumption of independence among all the variables could lead to a very simplistic approach for some problems. Some representative algorithms of this model are: the Population-Based Incremental Learning algorithm (PBIL) [5], the compact Genetic Algorithm (cGA) [18] and the most extended heuristic within this model, the Univariate Marginal Distribution Algorithm UMDA [26,31] and its continuous version, the Univariate Marginal Distribution Algorithm for Gaussian Models ($UMDA_g$) [21,23].

A slightly more sophisticated approach than the previous model is to consider the dependencies existing between two variables (pairwise or bivariate dependencies). This implies a good trade-off in terms of complexity and efficiency as, at most, one variable may depend on another one. To construct such graphical models, greedy approaches which add arcs to an initially disconnected graph are normally used. Some of the most frequently used heuristics within this model are MIMIC [7], COMIT [6] and BMDA [34]. All these approaches are available in both discrete and continuous versions.

The multivariate models are more flexible models since they do not constrain the number of dependencies among the variables. Their main disadvantage is that the computational cost of learning and sampling such models can be considerably high. An additional drawback is that, in order to construct an accurate model, the required population size should be bigger than for simpler models. Some examples of heuristics in the discrete case are EBNA [22] and BOA [33]. In the continuous case, most representative heuristics are $EMNA_{global}$ and EGNA [25].

For this work we have focused on two of the EDAs that have been more extensively used on continuous optimization: the univariate $UMDA_g$ and the more complex multivariate $EMNA_{global}$. No bivariate model has been included in this study since these two models have been the preferred choice in previous studies, the selected multivariate model can represent the same type of interactions than the bivariate approach and also with the aim of investing more effort in conducting a more extensive study in these two learning models instead of reducing the parameter values to include a third model.

2.2. $UMDA_g$

$UMDA_g$ assumes that the joint density function follows a n-dimensional normal distribution, which is factorized by a product of one-dimensional and independent normal densities. In every generation and for every variable, the UMDA_g carries out some statistical tests in order to find the density function that best fits the sampling of that variable. UMDA_g is a structure identification algorithm because the density components of the model to be learnt are identified via hypothesis tests. This estimation of parameters is carried out, after the densities are identified, by their maximum likelihood estimates. If all the univariate distributions are normal, then the two parameters to be estimated at each generation and for each variable are the mean, μ_i^l, and the standard deviation, σ_i^l. It is well known that their respective maximum likelihood estimates are:

$$\widehat{\mu_i^l} = \overline{X_i^l} = \frac{1}{N}\sum_{r=1}^{N} x_{i,r}^l \tag{1}$$

$$\widehat{\sigma_i^l} = \sqrt{\frac{1}{N}\sum_{r=1}^{N}\left(x_{i,r}^l - \overline{X_i^l}\right)^2} \tag{2}$$

2.3. $EMNA_{global}$

The Estimation of Multivariate Normal Algorithm (EMNA) is an approach based on the estimation of a multivariate normal density function. At each generation, the vector of means $\mu = (\mu_1, \ldots, \mu_n)$ and the variance–covariance matrix Σ whose elements are denoted by $\sigma_{i,j}^2$ with $i, j = 1, \ldots, n$, are estimated. Therefore, it requires the estimation of $2n + \binom{n-1}{2}$ parameters at each generation: n means, n variances and $\binom{n-1}{2}$ covariances. This can be done efficiently in a single pass through the population.

By its definition, the covariance matrix is always positive semi-definite but, because of the numerical errors caused by the finite precision of computers, sometimes the covariance matrix contains negative values. The problem with an ill-posed covariance matrix is that the sampling of new individuals becomes impossible. There are several solutions to repair the covariance matrix [13]. In this work, the most efficient method, the ECMR0 technique, has been selected for repairing the covariance matrix.

2.4. Distributed Estimation of Distribution Algorithms: dEDAs

In the distributed Evolutionary Algorithms (dEAs),[1] the whole population is distributed over multiple subpopulations and occasionally allows the migration or exchange of some individuals among the different islands. Therefore, each node executes an independent algorithm on an independent population. An important aspect of the performance of dEAs is the migration strategy. This is configured through different parameters [8]:

- Migration frequency: How often (number of generations) is information sent?
- Migration rate: How many individuals migrate each time?
- Information selection: What kind of information is exchanged?

[1] Also known as coarse-grained, multiple-deme or island models.

- Acceptance policy: How are the incoming and the local information combined?
- Migration topology: Which island sends information to which other?

Close scrutiny of migration parameters [37] has proved that, even though EAs with small populations risk being trapped in a local optimum, an appropriate migration strategy can avoid a suboptimal solution from dominating all the populations. A correct configuration can help to obtain better results with fewer evaluations, but configuring these optimal parameters is not a simple issue [3,30,42]. This appropriate strategy must be adjusted between the limits of a low interaction (which would practically imply the execution of N independent algorithms) and an excessive interaction (that would lead to the predominance of only one solution).

Regarding the information exchanged among islands, two possible alternatives are available: (i) the straightforward approach of selecting a pull of individuals that will be later sent to the consignees and (ii) the alternative of using the main characteristic of EDAs: the probabilistic models. These probabilistic models will be (or should be) able to represent the (in) dependencies among the variables, and, therefore, comprise more information than a group of individuals. This second approach opens a new challenge: how should the different probabilistic models be combined? In general, the combination of the resident model with the immigrant one can be formalized by the following rule [19]:

$$M'_R = \beta M_R \circ (1 - \beta) M_I \tag{3}$$

where $\circ$ represents the combination operator of two probabilistic models and β represents the influence of the immigrant model M_I on the resident model M_R and varies in the range [0,1]. An extended version of this formula for n immigrant models would be:

$$\begin{aligned} &M'_R = \beta_{I0} M_R \circ \beta_{I1} M_{I1} \circ \beta_{I2} M_{I2} \circ \ldots \circ \beta_{In} M_{In} \\ &\sum_i \beta_{Ii} = 1 \end{aligned} \tag{4}$$

A general approach for carrying out this combination is to apply a *mixture* model $M = \sum_i \beta_i M_i$ as a linear combination of simple distributions, where the β_i's satisfy that $\sum_i \beta_i = 1$ [11]. In this case, the population from the mixture model is created by sampling the individuals from each of the involved models. The β_i value defines the probability for selecting each model for the sampling of each new individual. This way, the models with higher β values, would have a higher probability of producing more individuals of the new population and would have more influence on the next inferred model.

The previous approach has the advantage of not depending on the inner details of the model and, therefore, can be applied to probabilistic models of different nature. A different strategy is to use a specific operator that takes into account the structure of the models to combine them appropriately. This method has the disadvantage that, with complex models, the combination is not trivial and, in some cases, can be very inefficient [20]. In the simple $UMDA_g$ models, a straightforward and efficient approach that can be carried out, named $UMDA_g$ combination method, is to combine the means and variances vector, i.e.:

$$\mu'_R = \sum_{i=1}^{n} \beta_i \mu_i, \quad \sigma'_R = \sqrt{\sum_{i=1}^{n} \beta_i^2 \sigma_i^2} \tag{5}$$

For this work we have introduced a new combination model called *uniform combination* for $UMDA_g$. In this method, instead of combining the models, each model component (mean and variance) is selected from a single model of the whole set of models. Each model has a probability β_i of being selected for each of the components of the new model. Therefore, the best models would contribute to more components than the worst ones. This approach is similar to the *mixture* model in the sense that each model is selected according to the β value. However, for this approach, the selected model is applied to a subset of dimensions for all the new individuals, whereas in the mixture case, the selected model was applied to all the dimensions of a single individual.

In order to compute the value of β, the strategy that has been followed in previous studies [10,20] is the so-called adaptative learning strategy. This method computes the β values based on the *quality* of the population associated to each model.

For n immigrants, the β value is defined as:

$$\beta_R = \frac{F_R}{F_R + \sum_j^n F_{I_j}}, \quad \beta_{I_i} = \frac{F_{I_i}}{F_R + \sum_j^n F_{I_j}} \tag{6}$$

where F_R represents the mean fitness value of the resident subpopulation and F_{I_i} represents the mean fitness value of the ith immigrant subpopulation. A conservative policy is also followed and only those models with a population of better quality than the resident model are considered for the computation. Therefore, if all the immigrants models have worse quality than the resident model, no combination is carried out.

3. Experimentation

For the experimentation, the benchmark from the workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems – A Scalability Test* held at the ISDA 2009 Conference has been considered. This benchmark defines 11 continuous optimization functions. The first 6 functions were originally proposed for the "*Special Session and Competition on Large Scale Global Optimization*" held at the CEC 2008 Congress [41]. The other 5 functions have been specially proposed for the Workshop of the ISDA 2009 Conference. These functions, presented in Table 1, have different degrees of difficulty and can scale to any dimension. Detailed information about the selected benchmark can be found at the web page of the organizers of the workshop.[2]

In order to analyze the effects of the migration strategies, several island configurations instances were compared against each other. The selection of the parameter values, such as the learning model, was based on the values used in previous studies with dEDAs [12,20] and was extended with additional values to obtain a wider view. Table 2 shows the different parameter values used throughout the experiments.

For the experimentation a full factorial design was chosen in order to conduct a complete study on the effect of each parameter on the response variable, as well as the effects of interactions between the parameters. Therefore, all possible combinations of the values across all the parameters were considered. This design has the disadvantage that it requires a higher number of runs but, in contrast, it can identify all the possible interactions between the parameters.

In order to make the results comparable with other algorithms, we have strictly followed the conditions imposed by the benchmark. Therefore, for each combination, 25 independent executions were carried out. The stopping criterion, as defined in the benchmark, was a fixed number of fitness evaluations (5000 times the dimension of the problem). The performance criterion (i.e. the response variable) is the distance (error) between the best individual found and the global optimum in terms of fitness value. Sequential versions of both the $UMDA_g$ and $EMNA_{global}$ algorithms were also executed with different population sizes (64, 128, 256, 512, 1024 and 2048) in order to have a baseline comparison.

4. Analysis of the results

For accomplishing the objectives proposed at Section 1, several analyses have been conducted from different perspectives.

4.1. Overall analysis

In order to conduct this study, a standard global measure was obtained for each configuration. First, the configurations were ranked on each function according to the procedure defined in the Friedman test. This way, the results on each function are defined in the same scale and can be averaged for obtaining the global rank measure.

The first approach that was considered was to use the Analysis of Variance (ANOVA) technique for analyzing the interactions between the parameters. ANOVA was developed by R.A. Fisher and has been applied to a vast array of different domains for data analysis. However, despite its widespread use, some key assumptions must be checked before applying ANOVA to the experimentation data: (i) the distributions of the residuals must follow a normal distribution and (ii) the variance of data in groups should be the same (equality of variances or homoscedasticity). For checking the first assumption, the Anderson–Darling, Shapiro–Wilk and Kolmogorov–Smirnov tests were carried out for each dimension of the global results obtaining for all tests a $p\text{-}value < 0.05$ and therefore rejecting the null hypothesis that the distributions of the residuals come from a normal distribution. The homocedasticity property was checked with both the Bartlett's and the Levene's tests obtaining a *p-value* less than $1e-4$ and therefore rejecting the assumption of homocedasticity on the results.

Due to these results, the Taguchi method [40] was selected as an alternative procedure for conducting the overall analysis. In this method, the concept of signal to noise ratio (SN ratio) is introduced for measuring the sensitivity of the quality characteristic being investigated in a controlled manner to those external influencing factors (noise factors) not under control. The aim of the experiment is to determine the *highest* possible SN ratio for the results since a high value of SN ratio implies that the signal is much higher than the random effects of the noise factors. From the quality point of view, there are three possible categories of quality characteristics: (i) smaller is better, (ii) nominal is best and (iii) bigger is better. The obtained results fall in the "smaller is better category" since the objective is to reduce the error between the best solution found and the global optimum. For this category, the SN ratio estimate is defined in Eq. (7) where n denotes the total number of instances and $y_1, y_2, \ldots, y_n$ the target values (the error to the best solution in this case).

$$SN = -10\log\left(\frac{1}{n}\sum_{t=1}^{n} y_t^2\right) \tag{7}$$

Figs. 1–3 display the main effects plot for the SN ratio results obtained for 50, 100 and 200 dimensions respectively. A main effect plot is a plot of the mean response values (the SN ratio for these graphs) at each level of a design parameter. This

[2] http://sci2s.ugr.es/programacion/workshop/Scalability.html.

Table 1
Benchmark functions.

Id	Name
*f*1	Shifted Sphere Function
*f*2	Shifted Schwefel's Problem 2.21
*f*3	Shifted Rosenbrock's Function
*f*4	Shifted Rastrigin's Function
*f*5	Shifted Griewank's Function
*f*6	Shifted Ackley's Function
*f*7	Schwefel's Problem 2.22
*f*8	Schwefel's Problem 1.2
*f*9	Extended f_{10}
*f*10	Bohachevsky
*f*11	Schaffer

Table 2
Parameters values.

Common parameter values	
Problem size	50, 100 and 200
Population size	512, 1024 and 2048
Learning model	$UMDA_g$ and $EMNA_{global}$
Selected inds. for learning	Best 50% of the population
#Islands	8 and 16
Topology	Ring2 and all-to-all (a2a)
Migration period	Migrate every 10, 20 and 40 generations
Acceptance policy	Best individuals from resident and Immigrants populations
Particular parameter values	
Inds. migration rate	10% and 20%
Inds. emigrants selection	Best or random individuals
Models combination	Mixture, UMDA combination and UMDA uniform

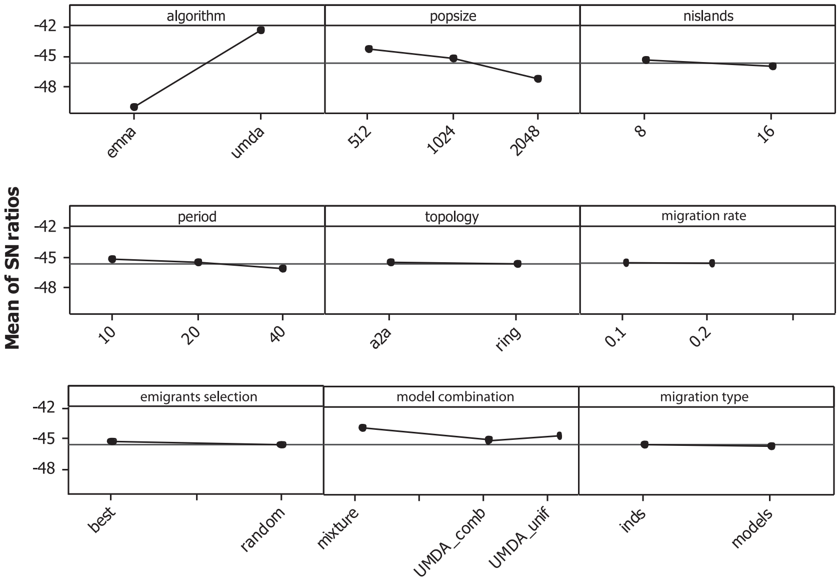

Fig. 1. Main effects plot for SN ratios on 50 dimensions.

plot can be used to compare the strength of the effects of the values of the parameter. From these results it can be observed that the most important parameter for characterizing the performance is the type of algorithm, having obtained the $UMDA_g$ model better SN values than the $EMNA_{global}$ approach. There are two main reasons that could explain this behavior: most of the functions (7 out of 11), although not linearly separable, can be easily optimized dimension by dimension, being the

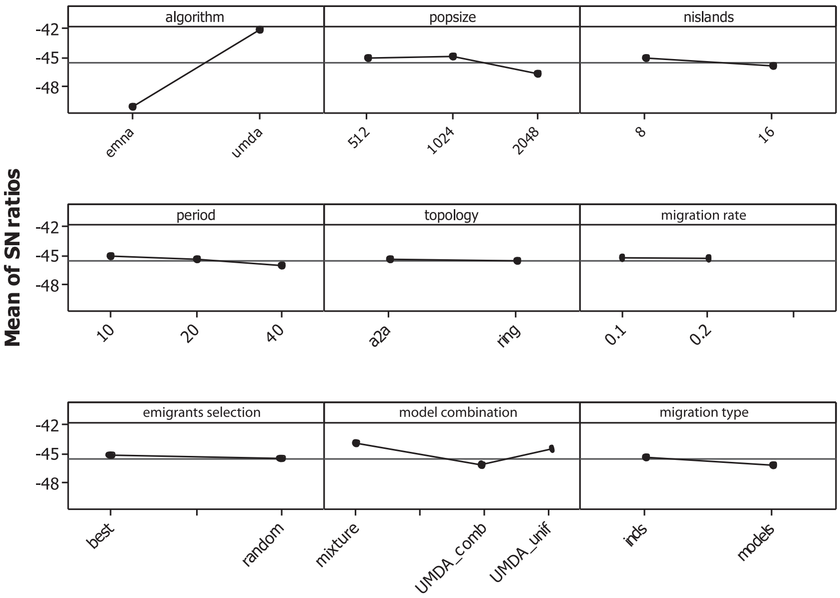

Fig. 2. Main effects plot for SN ratios on 100 dimensions.

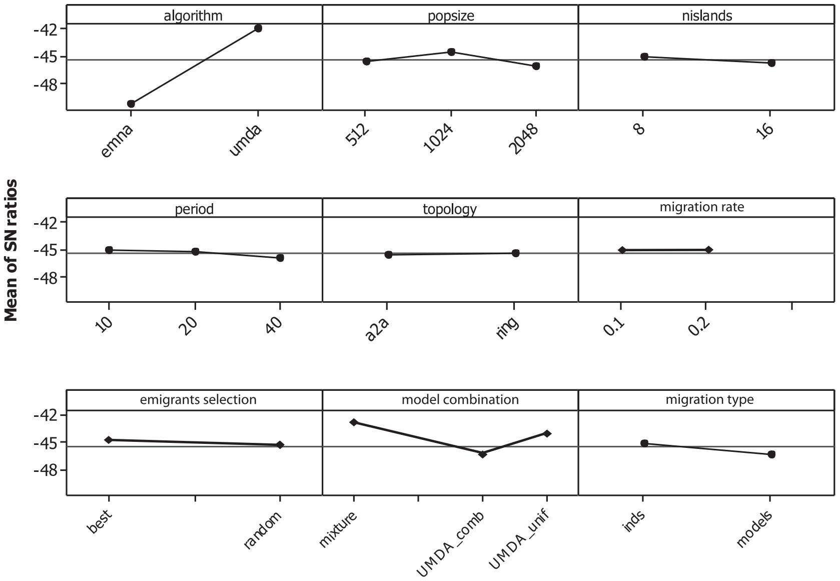

Fig. 3. Main effects plot for SN ratios on 200 dimensions.

UMDA$_g$ algorithm more suitable for this kind of functions since it does not manage any kind of interaction in its model. Second, the *EMNA*$_{global}$ algorithm needs to have a considerable amount of individuals to be able to capture all the dependencies among the variables [13]. However, in this benchmark the maximum number of evaluations is fixed and larger population sizes mean less number of iterations which heavily penalizes the final results. It must be taken into account that this constraint was imposed in the original benchmark as an example of the limitations that metaheuristics must deal with when solving real-world problems (the balance between execution time and solution quality).

The remaining parameters are not as decisive as the type of algorithm although some important patterns must be mentioned. Regarding the population size, 2048 individuals has obtained the worst values. In general, the EDAs tend to improve their results with larger population sizes (more information to build the model). However, due again to the number of evaluations constraint, a bigger population size implies a smaller number of iterations which, at the end, can severely affect the final values. In general, 1024 individuals has offered a good trade-off between the number of solutions for building a quality model and the maximum number of allowed iterations, although with the 50 dimensional functions, 512 individuals has obtained the best value. A similar trend has been observed with the number of islands. The smallest value, i.e., the one with the highest population size per island, has obtained the best signal to noise ratio values.

The next parameter that should be taken into account is the method for exchanging information. Regarding the two approaches for sending information, individuals or models, it can be mentioned that sending individuals has obtained better overall results being this difference bigger for the higher dimensional functions.

From the two remaining common parameters, it seems that there is no significant difference among the two topology values explored and that a more frequent exchange of information has been more beneficial for obtaining the best values. Concerning the particular parameter values for sending individuals, the best selection of emigrants has obtained a small improvement over the random selection and, apparently, both selection rate values have obtained similar results. For the analysis of the different methods for combining models, only the results from the $UMDA_g$ algorithms have been considered since for the $EMNA_{global}$ configurations only one method for combining models was proposed. From these results it is clear that the mixture model obtains the best results followed by the new proposed UMDA uniform model and having obtained the UMDA combination model the worst results among the three dimensions.

An analysis of the interactions between the parameters, measured with the SN ratio, was also conducted. Figs. 4–6 display the interactions plots between the parameters of the configurations based on sending individuals in 50, 100 and 200 dimensions, respectively. In the same way, Figs. 7–9 represent the interactions of the configurations based on sending models. An interaction plot is a powerful graphical tool which displays the mean response of two parameters (the SN ratio in this case) at all the possible combination of their values. For improving the clarity of the diagrams, only the parameters that have presented any interaction with any other parameter have been displayed in these graphs.

Several interactions have been identified in the configurations based on sending individuals. One of the most remarkable interactions is the relationship between the topology and the criterion for selecting the immigrants, where the worst results are obtained with the all-to-all topology and the selection of the best individuals. This seems logical since, in this combination, the diversity of the algorithm is quickly reduced in the first generations. A similar scenario arises between the population size and the topology parameters, where the performance of the all-to-all topology is heavily penalized when connected with the smallest population size, especially with the highest dimensional functions. Another interesting relationship is the one between the population size and the number of islands in the highest dimensional functions. With 512 individuals, the increment of the number of islands from 8 to 16 leaves a population of 32 individuals per island, which is clearly not enough for inducing a good model of the problem.

Several interactions have also arisen with the configurations that exchange models. As for the sending individuals configurations, the increment from 8 to 16 islands considerably reduces the performance of the configurations that use the smallest population size. Regarding the methods for combining the models, it seems that the UMDA combination method, in contrast to the other alternatives, obtains better results with the highest population sizes, lower migration frequencies and the lowest topology degree approach, i.e., characteristics that tend to increment the global diversity of the algorithm.

The previous studies have provided a broad view of the overall behavior of all the configurations. With the purpose of offering an alternative view of the relationships between the parameters and the algorithm performance, the best and worst

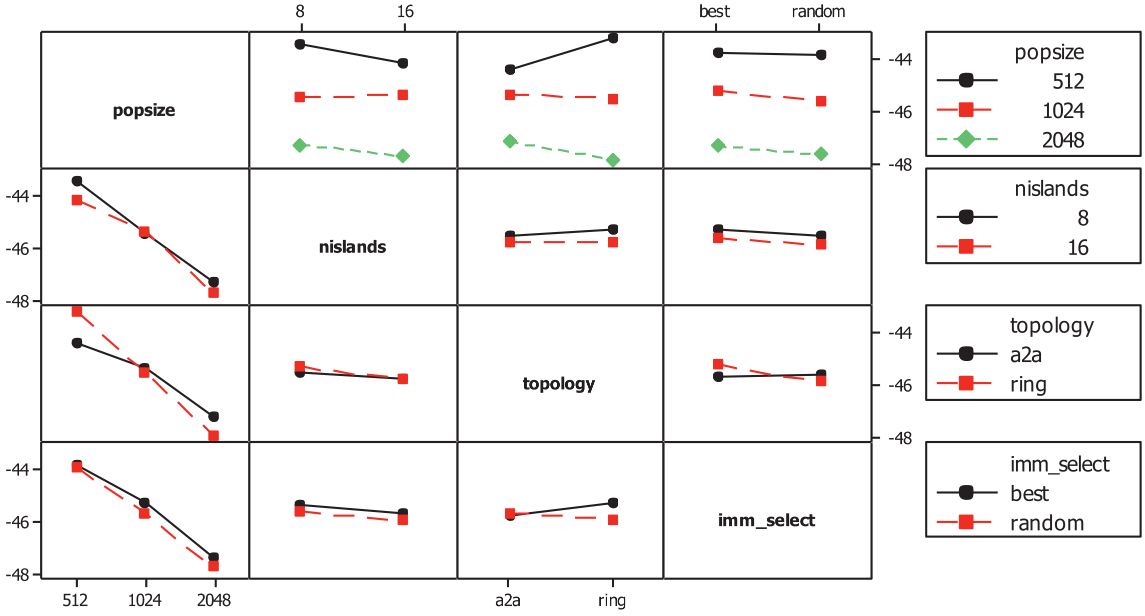

Fig. 4. Sending individuals interactions plot for SN ratios on 50 dimensions.

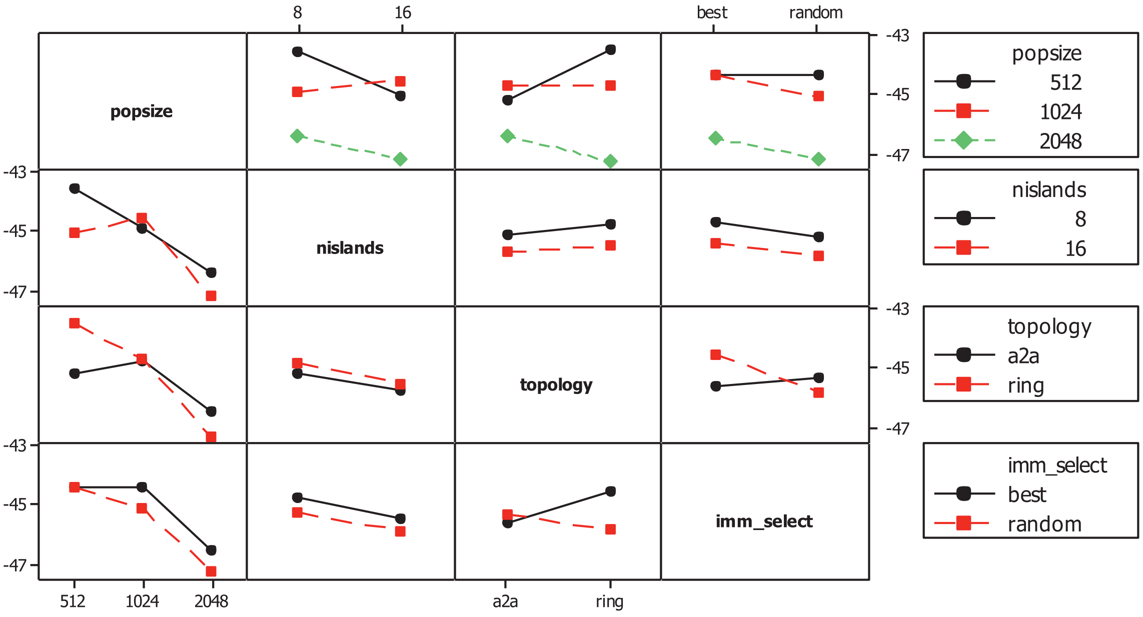

Fig. 5. Sending individuals interactions plot for SN ratios on 100 dimensions.

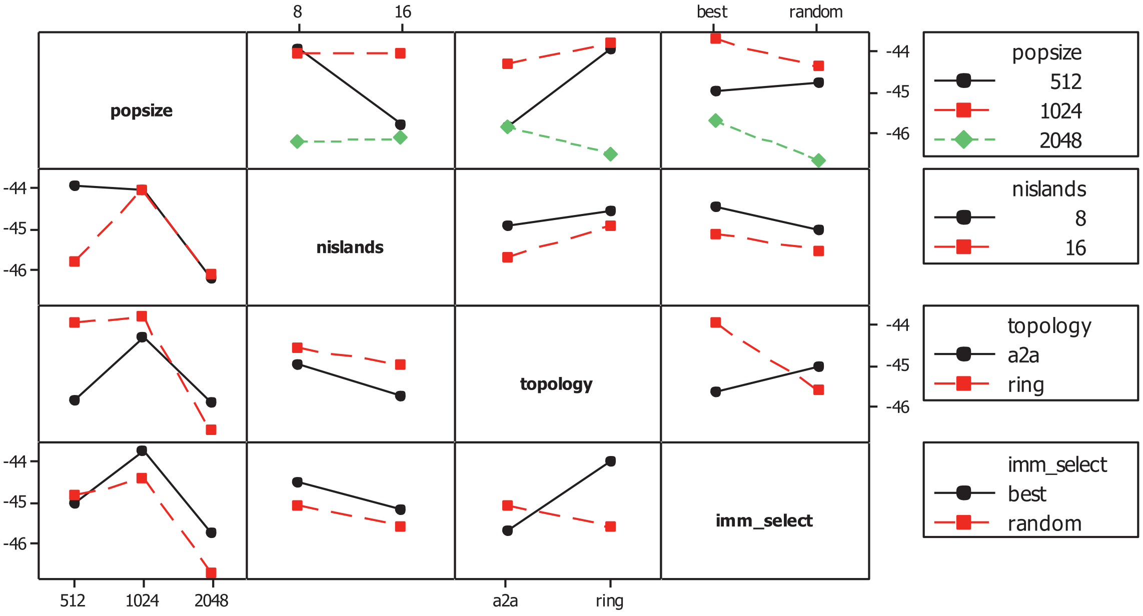

Fig. 6. Sending individuals interaction plot for SN ratios on 200 dimensions.

configurations have been analyzed independently from the remaining configurations. Moreover, the sequential algorithms have been included in this study in order to compare the performance of the distributed configurations. Tables 3–5, display a ranking of the configurations on 50, 100 and 200 dimensions, respectively, according to the global average rank. In these tables only the best and worst 5% of the configurations, as well as the sequential configurations, are displayed[3].

From these tables it is clear that the best results have been obtained with distributed configurations, being the sequential algorithms placed around positions near the middle of the table. To better analyze the influence of the parameters, Figs. 10–12 display the parallel coordinates graph of the distribution of the values of the parameters of the best configurations on 50, 100 and 200 dimensions respectively and Figs. 13–15 the equivalent graphs for the worst configurations. Parallel

[3] The complete 444 results can be accessed in the following URL http://cajalbbp.cesvima.upm.es/storage.

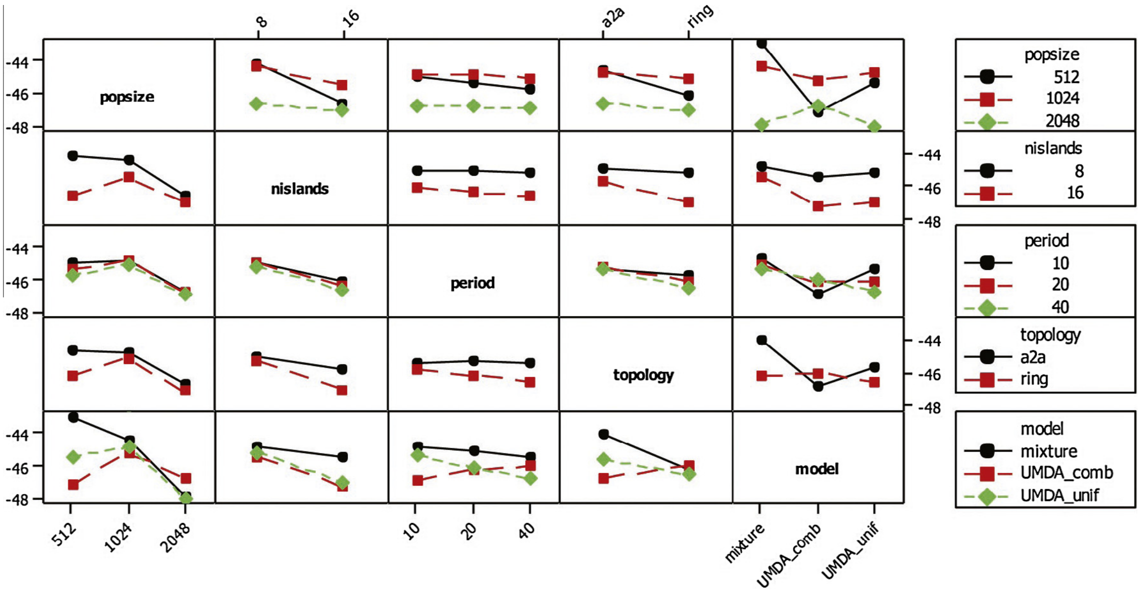

Fig. 7. Sending models interactions plot for SN ratios on 50 dimensions.

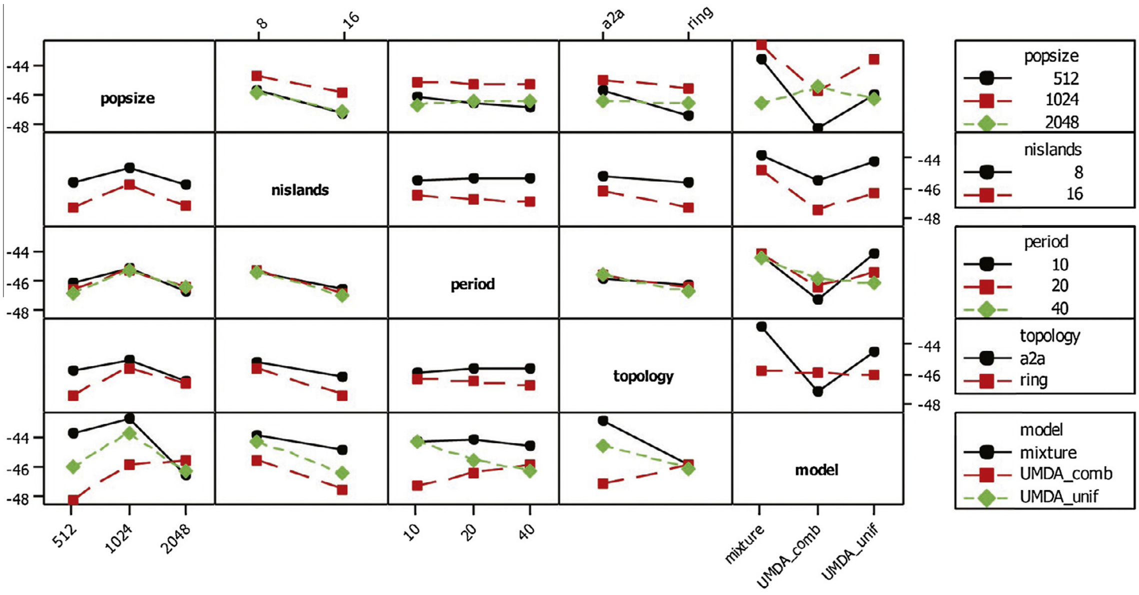

Fig. 8. Sending models interactions plot for SN ratios on 100 dimensions.

coordinates is a useful technique which has been successfully used to represent high dimensional data as polylines in two dimensions. More recently, it has been used to capture the underlying interactions between the parameters of a Particle Swarm Optimization algorithm [15]. In this paper, the first eight axes represent the parameters of the configurations, whereas the ninth one represents the Friedman average ranking mentioned before. Only the best and worst 20% of the solutions are displayed on these graphs. The best/worst 5% of the solutions have been highlighted with a variety of different colors while the remaining 15% has been marked in light gray. The global best and worst configurations have been depicted with a dashed line. A small deviation, similar to a jitter effect, has been applied to each configuration parameter value to distinguish better the configurations.

As happened in the first analysis, the $UMDA_g$ learning model has been the most decisive parameter: all the best results have been achieved with this learning model, whereas the $EMNA_{global}$ model has been the common value among the worst algorithms. Regarding the information exchange methods, it can be observed that sending individuals obtains the best

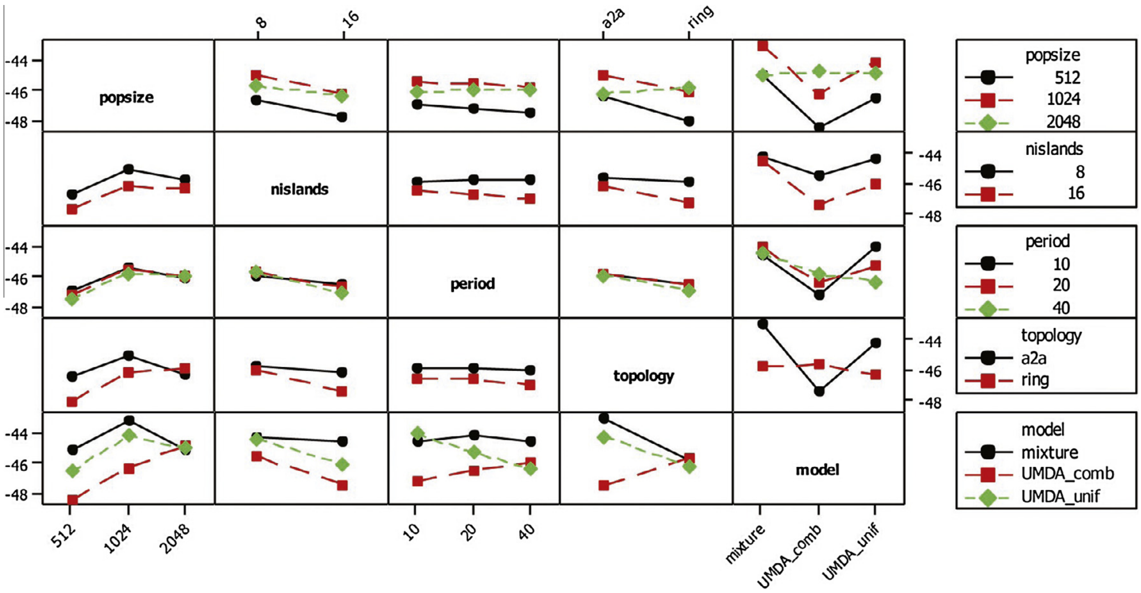

Fig. 9. Sending models interaction plot for SN ratios on 200 dimensions.

overall results and that this characteristic is present in almost all the best 5% of the configurations. However, the opposite is not true, since several configurations that send individuals are among the worst 5%. This fact explains why the difference between these two approaches has not been very significant in the first study. Sending individuals has demonstrated to be the best choice for this benchmark but needs to be properly configured since, without the proper selection of the remaining parameters, this value would not determine the performance of the configuration. A similar situation happens with the topology parameter, where the ring topology has been selected by almost the whole set of the best configurations. For 100 and 200 dimensions the situation is quite similar, confirming the fact that the previous conclusions are stable and can be generalized to a higher number of dimensions.

4.2. Comparing the performance on each function

For this study, it was planned to include the characteristics of the problems in the data set to see the impact of these characteristics in the solution quality. These characteristics were extracted from the literature and are represented in Table 6.

However, after conducting an analysis of the main effects and interaction graphs for each function, it was observed that the behavior of similar functions, according to this table, was completely different according to the graphs. For example, for functions $f2$ and $f8$, which both are unimodal, shifted and not separable, the corresponding main effects plots have little in common as it can be shown in Figs. 16 and 17 for 100 dimensions. Moreover, some functions that have opposite characteristics in Table 6, such as $f1$ and $f5$, have demonstrated to behave almost identically according to the main effects graphs as represented in Figs. 18 and 19. Therefore, this type of analysis could not be carried out.

By contrast, the analysis of the main effects graphs has identified two groups of functions with similar behaviors: one made up of functions $f1$ and $f5$ and a second one made up of functions $f7$, $f9$, $f10$ and $f11$. The remaining functions have presented a particular behavior, significantly different to be included in any other group.

Finally, it is worth mentioning that only in functions $f4$ and $f8$ the exchange of models obtained a better SN ratio than the exchange of individuals. In order to provide more insight about this behavior, a study for comparing both information exchange methods was carried out.

This study consisted in analyzing the performance *on each function* of the configurations based on sending individuals against the equivalent ones based on sending models. For this task, all the configurations were grouped according to their information exchange model: 4 groups for sending individuals and 3 for sending models with 61 configurations per group. Then, the average rank was obtained for each group (61 values per group). Tables 7–9 present the results in 50, 100 and 200 dimensions. For each function, the best average rank is highlighted on both tables.

From these results, it can be seen that, in 9 out of 11 functions, the groups based on sending individuals obtained a superior average rank than the ones based on sending models. Only on two functions, $f4$ and $f8$, the sending models configurations based on the $UMDA_g$ combination method obtained a better average rank than the sending individuals configurations. In order to explain this behavior, it is convenient to understand that the exchange and combination of models has the objective of improving the accuracy and reducing the noise of the current model by its combination with the models sent from the other islands. However, when exchanging models instead of individuals, the best solutions that

Table 3
Average rankings 50 dimensions.

Alg	Size	#Islands	Period	Topology	Rate	Emm. selec	Model	Avg. ranking
Best 5%								
umda	512	8	10	ring	0.2	best	–	41.09
umda	512	8	10	ring	0.1	best	–	42.32
umda	512	16	10	ring	0.2	best	–	44.14
umda	512	16	10	ring	0.1	best	–	46.91
umda	512	8	20	ring	0.2	best	–	47.14
umda	512	8	20	ring	0.1	best	–	49.64
umda	512	16	10	ring	0.2	random	–	50.64
umda	512	16	20	ring	0.1	best	–	51.59
umda	512	16	20	ring	0.2	best	–	52.00
umda	512	8	10	ring	0.2	random	–	53.09
umda	512	16	10	ring	0.1	random	–	54.64
umda	512	8	20	ring	0.2	random	–	57.18
umda	512	8	10	ring	0.1	random	–	61.18
umda	512	16	20	ring	0.2	random	–	61.23
umda	512	8	20	ring	0.1	random	–	61.36
umda	512	8	40	ring	0.1	random	–	61.55
umda	512	16	20	ring	0.1	random	–	62.68
umda	512	8	40	ring	0.2	best	–	67.68
umda	512	8	40	ring	0.2	random	–	68.41
Sequential configurations								
umda	512							110.50
umda	256							126.23
umda	1024							142.23
umda	128							148.05
umda	2048							208.82
umda	64							212.64
emna	2048							236.91
emna	1024							258.09
emna	512							316.91
emna	256							360.82
emna	64							364.64
emna	128							371.73
Worst 5%								
emna	512	8	40	ring	0.1	random	–	377.82
emna	1024	16	10	ring	0.1	random	–	380.45
emna	1024	16	20	ring	0.2	random	–	381.00
emna	512	8	40	a2a	–	–	mixture	383.45
emna	512	8	20	a2a	–	–	mixture	385.36
emna	512	8	40	ring	–	–	mixture	387.00
emna	1024	16	20	ring	0.1	random	–	388.36
emna	1024	16	40	ring	0.2	random	–	388.82
emna	512	16	10	ring	–	–	mixture	389.09
emna	512	8	20	ring	–	–	mixture	389.55
emna	1024	16	40	ring	0.1	random	–	392.27
emna	512	8	10	a2a	–	–	mixture	392.91
emna	1024	16	10	a2a	–	–	mixture	394.82
emna	1024	16	20	a2a	–	–	mixture	394.91
emna	1024	16	40	a2a	–	–	mixture	396.91
emna	1024	16	20	ring	–	–	mixture	397.18
emna	1024	16	10	ring	–	–	mixture	398.55
emna	512	8	10	ring	–	–	mixture	398.91
emna	1024	16	40	ring	–	–	mixture	399.00

the other islands have found are not available to the current island. As other studies have proven [24], the use of elitism is normally beneficial for the behavior of EDAs. For this reason, the combination of models offers a poorer performance on most of the functions.

Therefore, for the purpose of clarifying the behavior of the exchange of models in functions $f4$ and $f8$, the best sequential EDA was executed with and without elitism on the 50 dimensional functions.[4] Table 10 displays the average error of the 25 executions and highlights if the differences between both versions (with and without elitism) are significant according to the Wilcoxon signed-rank test. In this table it is shown that the absence of elitism obtains significantly better results in functions $f4$, $f6$ and $f8$. Furthermore, the results obtained in functions $f4$ and $f8$ by the sequential EDA

[4] Only in this dimension since the behavior is similar on the other dimensions.

Table 4
Average rankings 100 dimensions.

Alg	Size	#Islands	Period	Topology	Rate	Emm. selec	Model	Avg. ranking
Best 5%								
umda	512	8	20	ring	0.1	best	–	53.18
umda	512	8	20	ring	0.2	best	–	53.32
umda	512	8	10	ring	0.2	best	–	53.95
umda	512	8	10	ring	0.1	best	–	55.73
umda	512	8	10	ring	0.2	random	–	58.23
umda	1024	16	10	ring	0.2	best	–	60.77
umda	1024	16	10	ring	0.1	best	–	61.09
umda	512	8	40	ring	0.1	best	–	63.95
umda	512	8	40	ring	0.1	random	–	65.59
umda	1024	8	20	ring	0.2	best	–	66.23
umda	1024	16	20	ring	0.2	best	–	66.73
umda	512	16	10	ring	0.1	best	–	66.95
umda	1024	8	10	ring	0.1	best	–	66.95
umda	512	16	10	ring	0.2	best	–	67.32
umda	1024	16	20	ring	0.1	best	–	67.50
umda	512	8	40	ring	0.2	best	–	67.77
umda	1024	8	10	ring	0.2	best	–	68.82
umda	512	8	10	ring	0.1	random	–	71.00
umda	512	16	20	ring	0.1	best	–	71.00
Sequential configurations								
umda	512							113.68
umda	1024							129.77
umda	256							147.50
umda	128							166.50
umda	2048							171.05
umda	64							226.91
emna	2048							289.27
emna	1024							312.64
emna	256							330.64
emna	512							339.27
emna	64							363.45
emna	128							391.27
Worst 5%								
emna	1024	8	40	ring	0.1	random	–	411.36
emna	1024	8	20	ring	–	–	mixture	413.45
emna	1024	8	40	ring	–	–	mixture	415.09
emna	2048	16	40	ring	0.1	best	–	415.55
emna	1024	8	10	a2a	–	–	mixture	416.27
emna	1024	8	20	a2a	–	–	mixture	417.09
emna	1024	8	40	a2a	–	–	mixture	417.36
emna	2048	16	10	ring	0.1	random	–	418.27
emna	2048	16	20	ring	0.2	random	–	418.36
emna	1024	8	10	ring	–	–	mixture	418.82
emna	2048	16	40	ring	0.2	random	–	423.36
emna	2048	16	20	ring	0.1	random	–	424.55
emna	2048	16	10	a2a	–	–	mixture	425.00
emna	2048	16	20	a2a	–	–	mixture	426.18
emna	2048	16	40	ring	–	–	mixture	426.18
emna	2048	16	40	ring	0.1	random	–	427.91
emna	2048	16	10	ring	–	–	mixture	428.36
emna	2048	16	40	a2a	–	–	mixture	428.45
emna	2048	16	20	ring	–	–	mixture	429.55

are better than the obtained by the best distributed algorithm as it will be shown in the following study. It seems that, for these functions, it is better to improve the general diversity of the population (and therefore, the exploratory ability of the algorithm) in order to reach better regions of the search space. In *f*6, where both versions have almost converged to the final optimum, it seems that the greater diversity of the distributed model (against the sequential version) is enough for dealing with this characteristic and the non-elitist sequential EDA does not improve the results of the distributed versions.

Within the individuals configurations, sending the best 10% individuals achieves the best overall performance in most of the functions for both $UMDA_g$ and $EMNA_{global}$ models. On the other hand, sending a 20% of randomly chosen individuals obtained the worst results. Regarding the different methods for combining the models, it must be taken into account that

Table 5
Average rankings 200 dimensions.

Alg	Size	#Islands	Period	Topology	Rate	Emm. selec	Model	Avg. ranking
Best 5%								
umda	1024	8	20	ring	0.1	best	–	55.45
umda	1024	8	20	ring	0.2	best	–	56.05
umda	1024	8	10	ring	0.2	best	–	57.77
umda	1024	16	10	ring	0.2	best	–	58.23
umda	1024	16	10	ring	0.1	best	–	58.68
umda	512	8	20	ring	0.1	best	–	58.77
umda	1024	8	10	ring	0.1	best	–	59.91
umda	1024	16	10	ring	0.2	random	–	60.45
umda	1024	16	20	ring	0.2	best	–	61.59
umda	1024	16	20	ring	0.1	best	–	62.82
umda	512	8	10	ring	0.2	random	–	63.09
umda	512	8	10	ring	0.2	best	–	63.91
umda	512	8	20	ring	0.2	best	–	65.23
umda	512	8	40	ring	0.1	best	–	66.18
umda	1024	8	10	ring	0.2	random	–	66.27
umda	512	8	10	ring	0.1	best	–	66.77
umda	1024	8	40	ring	0.2	best	–	66.77
umda	512	8	10	ring	0.1	random	–	66.91
umda	512	8	20	ring	0.2	random	–	67.00
Sequential configurations								
umda	1024							121.18
umda	512							129.55
umda	256							154.50
umda	2048							163.32
umda	128							196.55
umda	64							234.00
emna	2048							304.82
emna	1024							317.09
emna	512							320.55
emna	64							364.14
emna	128							390.86
emna	256							422.14
Worst 5%								
emna	1024	8	40	a2a	–	–	mixture	413.59
emna	2048	16	20	ring	0.1	random	–	413.77
emna	1024	8	40	ring	–	–	mixture	414.14
emna	1024	8	20	ring	–	–	mixture	414.95
emna	1024	8	10	a2a	–	–	mixture	415.68
emna	2048	8	40	r-	–	mixture	420.77	
emna	2048	16	10	a2a	–	–	mixture	422.68
emna	2048	16	40	a2a	–	–	mixture	423.50
emna	2048	16	20	a2a	–	–	mixture	424.05
emna	2048	16	40	ring	–	–	mixture	424.59
emna	2048	16	20	ring	–	–	mixture	424.59
emna	2048	16	10	ring	–	–	mixture	425.23
emna	2048	8	40	ring	0.2	random	–	430.23
emna	2048	8	10	a2a	–	–	mixture	434.23
emna	2048	8	40	a2a	–	–	mixture	434.41
emna	2048	8	20	a2a	–	–	mixture	434.50
emna	2048	8	20	ring	–	–	mixture	434.86
emna	2048	8	10	ring	–	–	mixture	434.95
emna	2048	8	40	ring	–	–	mixture	435.14

within the $UMDA_g$ configurations, the mixture model obtains the best results in most of the functions followed by the proposed uniform approach, being both models better than the combination method.

4.3. Comparing the best configurations

A comparison of the best configurations for both type of algorithms and the different methods for exchanging information was conducted. The best sequential configuration for each algorithm was also included in this study along with the IPOP-CMA-ES algorithm, a metaheuristic that follows a strategy similar to that of the $EMNA_{global}$ since it learns a covariance matrix and uses it (along with more information) to sample the individuals of the new population. This algorithm was the winner of the *Special Session on Real-Parameter Optimization* of the IEEE CEC 2005

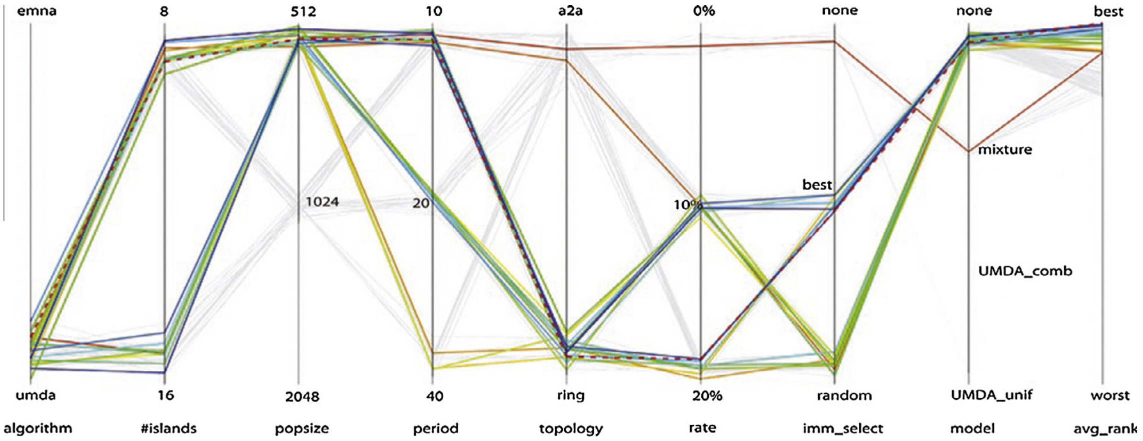

Fig. 10. Parameter values of the best configurations on 50 dimensions.

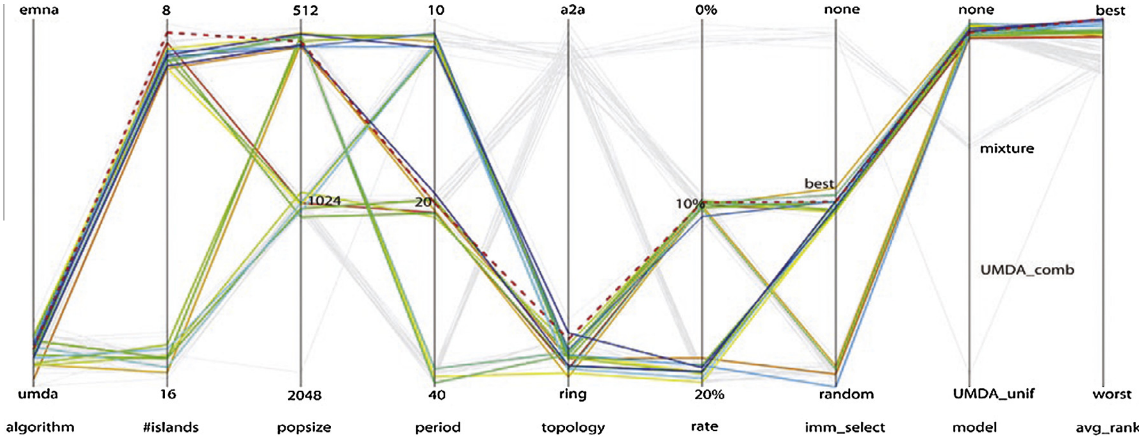

Fig. 11. Parameter values of the best configurations on 100 dimensions.

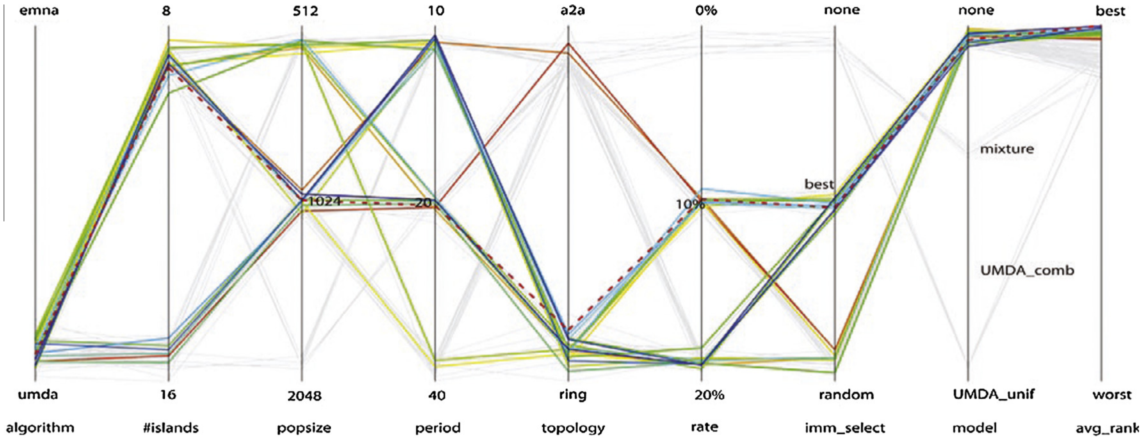

Fig. 12. Parameter values of the best configurations on 200 dimensions.

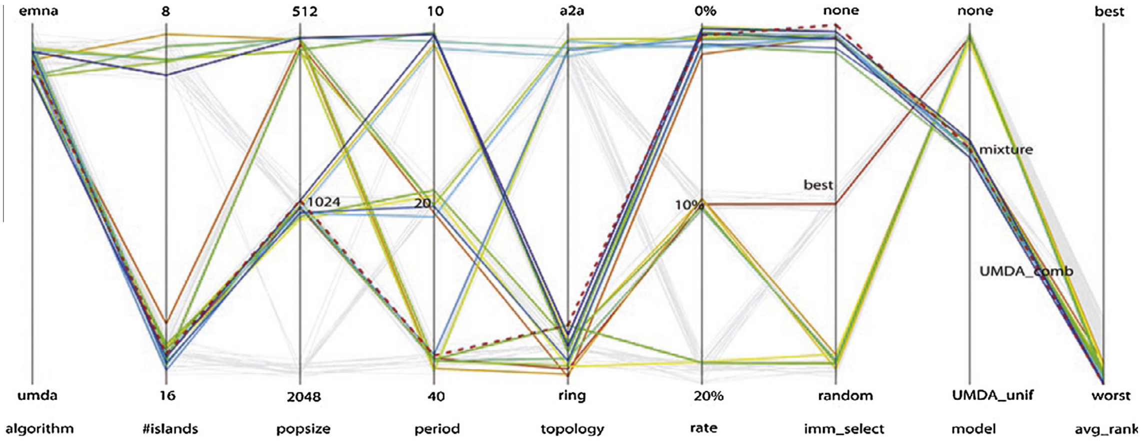

Fig. 13. Parameter values of the worst configurations on 50 dimensions.

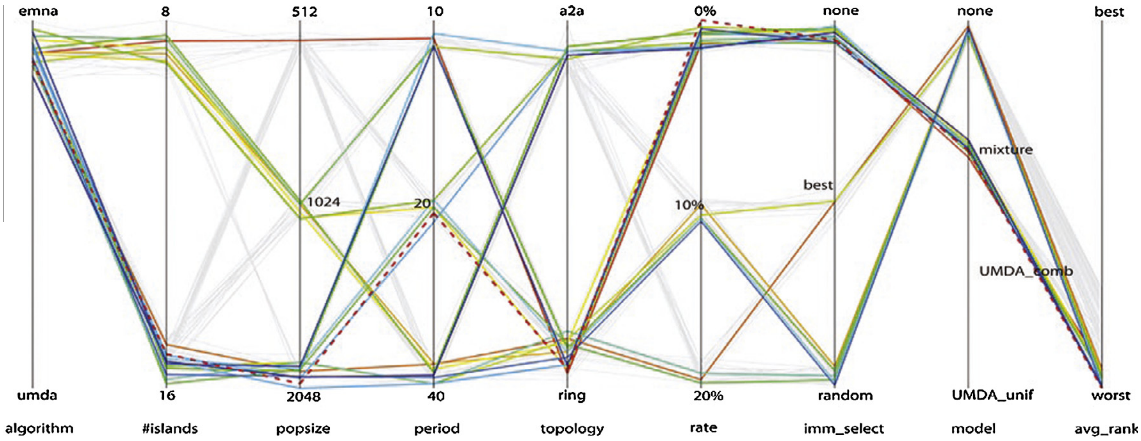

Fig. 14. Parameter values of the worst configurations on 100 dimensions.

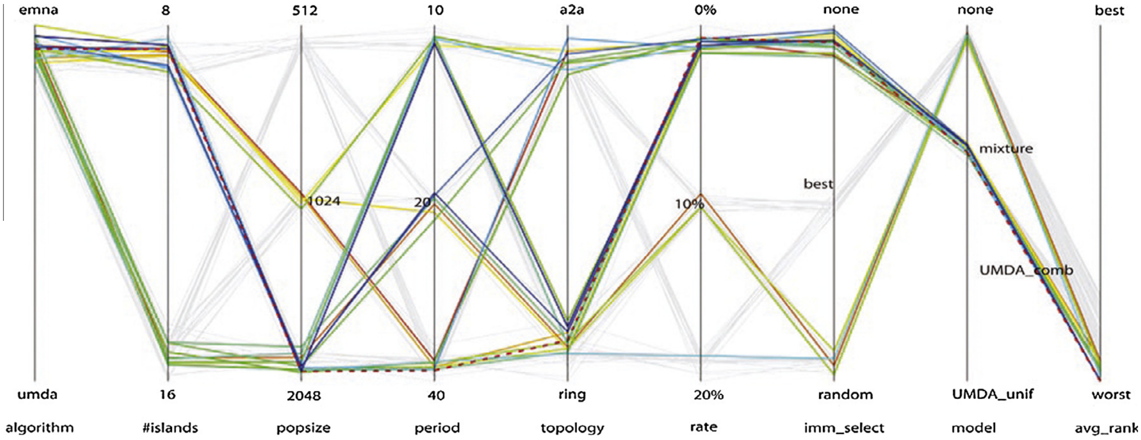

Fig. 15. Parameter values of the worst configurations on 200 dimensions.

Table 6
Properties of the functions of the benchmark.

Function	Unimodal/multimodal	Shifted	Separable	Easily optimized dimension by dimension
*f*1	U	Y	Y	Y
*f*2	U	Y	N	N
*f*3	M	Y	N	N
*f*4	M	Y	Y	Y
*f*5	M	Y	N	N
*f*6	M	Y	Y	Y
*f*7	U	Y	Y	Y
*f*8	U	Y	N	N
*f*9	U	Y	N	N
*f*10	U	Y	N	Y
*f*11	U	Y	N	N

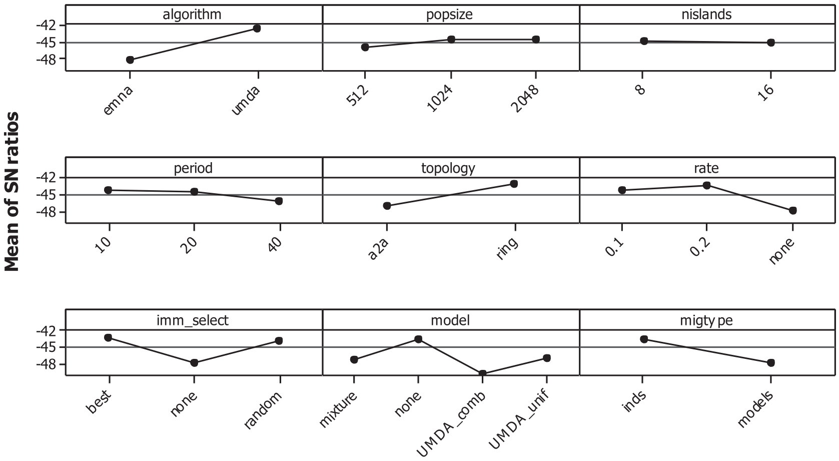

Fig. 16. Main effects plot for SN ratios on 100 dimensions for the *f*2 function.

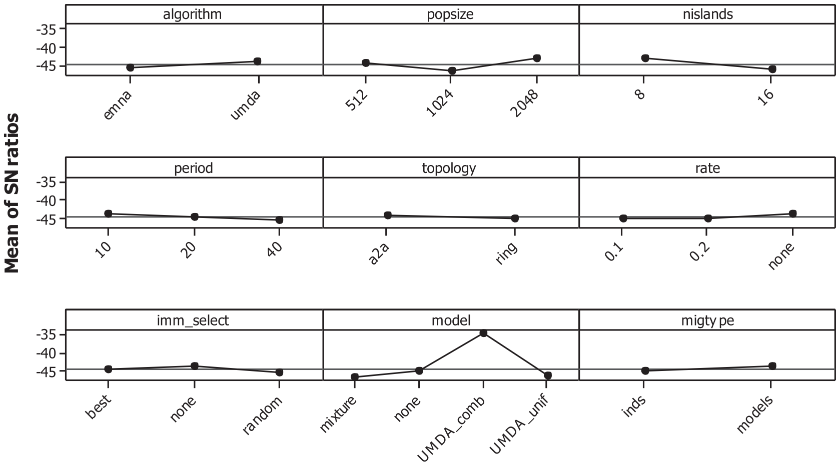

Fig. 17. Main effects plot for SN ratios on 100 dimensions for the *f*8 function.

conference [39] and was proposed by the organizers of the benchmark as a reference algorithm for comparing the results.

Table 11 displays the average error of the 25 executions on each function. As shown in the table, the best results on most of the functions are obtained by the distributed configuration that uses the $UMDA_g$ algorithm and exchanges individuals.

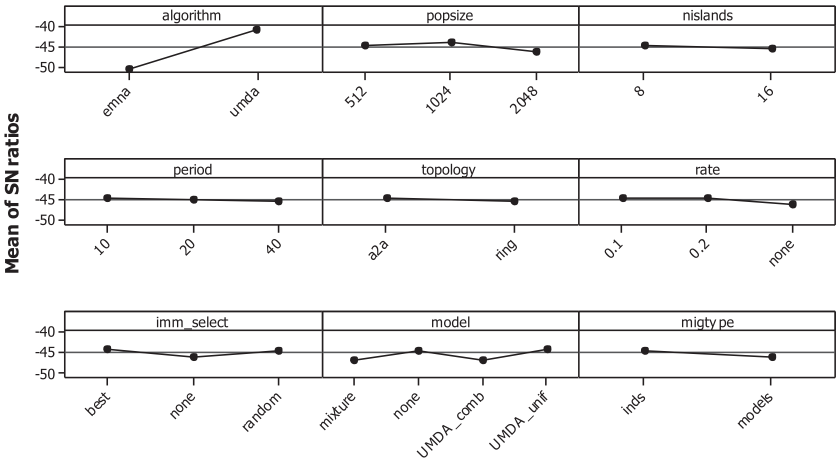

Fig. 18. Main effects plot for SN ratios on 100 dimensions for the *f*1 function.

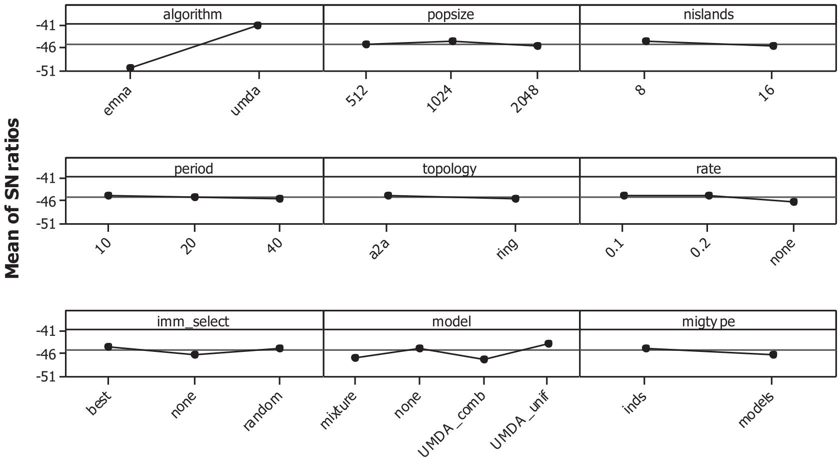

Fig. 19. Main effects plot for SN ratios on 100 dimensions for the *f*5 function.

Similarly to what happened in the previous analysis, the performance of the $EMNA_{global}$ configurations is significantly worse than the $UMDA_g$ ones. However, with the *f*8 function, both the sequential and the distributed exchanging models configuration obtain better results than the $UMDA_g$ ones. As shown on its function definition (Eq. (8)), the first components of the solution have more influence in the fitness function than the remaining components. With an appropriate population size, the $EMNA_{global}$ algorithm is able to detect this dependency between the variables and focus the search on the first dimensions of the solutions.

$$f8(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2, \quad z = x - o \tag{8}$$

o represents the global optimum

An interesting pattern that can be seen in this table is that, for the configurations that exchange models, the population size is larger for those based on sending individuals and the selected topology in both cases is the all-2-all method. It seems that, for most of the functions, it is better to have a larger population size for improving the quality of the induced model and share it with the rest of the islands to accelerate the diffusion of good patterns.

The performance of the sequential configurations is, in general terms, worse than their equivalent distributed configurations. However, in some functions (*f*9 and *f*11 for $UMDA_g$ and *f*8 for the $EMNA_{global}$) the concentration of the individuals helps to improve the results.

In order to provide a proper statistical validation of the results, the procedure described in [16,17] was followed, where the distribution of the results was first compared with the Friedman test in order to detect significant differences.

Table 7
Average ranking per function on 50 dimensions functions.

Function	0.1-best	0.1-random	0.2-best	0.2-random	Mixture	UMDA_comb	UMDA_unif
UMDA							
*f*1	**84.28**	89.53	93.32	97.04	135.03	220.03	166.44
*f*2	128.44	**127.17**	135.00	133.81	178.89	306.22	255.25
*f*3	76.69	**68.92**	94.17	93.53	148.92	224.50	181.67
*f*4	138.81	158.61	132.31	153.64	141.14	**97.86**	138.36
*f*5	**82.04**	92.74	91.06	103.40	133.08	226.72	162.49
*f*6	**88.71**	91.67	93.93	96.68	141.40	204.74	168.46
*f*7	107.94	120.94	**106.25**	117.17	147.51	136.00	149.68
*f*8	236.92	246.39	238.97	241.67	242.39	**109.11**	234.97
*f*9	117.28	120.79	**110.04**	119.10	150.11	133.50	134.68
*f*10	**109.81**	115.40	108.64	112.68	138.24	153.50	147.82
*f*11	115.19	119.86	**109.69**	117.61	153.86	132.97	136.31
EMNA							
*f*1	333.22	339.36	339.72	344.17	355.86		
*f*2	249.89	259.69	251.33	254.36	317.94		
*f*3	332.61	336.94	336.53	340.94	362.58		
*f*4	312.58	336.72	308.75	328.78	350.44		
*f*5	329.39	337.39	339.81	343.08	356.81		
*f*6	323.28	341.06	329.33	340.42	378.33		
*f*7	335.86	340.42	337.19	339.11	359.92		
*f*8	195.56	208.92	195.94	205.08	242.08		
*f*9	334.39	343.11	335.81	340.28	358.92		
*f*10	332.67	341.53	335.64	341.11	360.97		
*f*11	334.14	341.97	337.17	340.28	358.94		

Table 8
Average ranking per function on 100 dimensions functions.

Function	0.1-best	0.1-random	0.2-best	0.2-random	Mixture	UMDA_comb	UMDA_unif
UMDA							
*f*1	85.46	90.33	**81.07**	88.53	142.44	225.60	175.54
*f*2	123.33	**111.33**	124.31	120.39	163.06	317.31	247.83
*f*3	**74.19**	69.61	93.64	93.92	146.25	232.69	183.36
*f*4	129.56	146.44	125.03	143.00	134.58	**100.53**	131.33
*f*5	**84.22**	94.65	90.57	95.58	133.67	250.50	165.22
*f*6	84.22	**83.32**	98.46	92.96	141.46	219.81	165.78
*f*7	103.56	109.17	**102.28**	107.03	134.51	182.44	146.54
*f*8	169.81	209.58	176.06	197.00	214.58	**59.78**	213.28
*f*9	**105.53**	110.86	108.33	110.76	160.33	160.94	128.76
*f*10	100.64	**96.86**	107.51	107.15	151.56	176.64	145.42
*f*11	**106.57**	109.83	107.89	109.49	160.61	162.67	128.47
EMNA							
*f*1	325.17	342.33	324.03	333.50	384.00		
*f*2	253.75	271.67	250.00	259.56	355.47		
*f*3	324.25	340.61	321.31	332.47	385.69		
*f*4	313.11	347.06	310.08	330.75	386.53		
*f*5	319.08	336.56	315.75	327.25	384.94		
*f*6	320.28	344.75	319.83	335.00	392.14		
*f*7	324.81	343.64	323.64	333.67	386.72		
*f*8	242.44	277.22	237.06	258.11	343.08		
*f*9	323.50	345.58	322.25	335.89	385.25		
*f*10	323.72	345.28	321.53	334.78	386.92		
*f*11	324.31	346.22	322.03	334.92	385.00		

A value of 103.59 was obtained for the chi-squared statistic, which corresponds with a *p-value* of 4.45E−20, confirming the existence of significant differences between the results. Then, two post hoc methods (Holm and Hochberg) were used to obtain the adjusted *p*-values for each comparison between the control algorithm (the distributed $UMDA_g$ based on sending individuals).

The results of these tests are reported in Table 12, and show, for all of them, that there is statistical evidence to state that the dEDA algorithm based on sending individuals is significantly better than any of the EDA algorithms considered in the

Table 9
Average ranking per function on 200 dimensions functions.

Function	0.1-best	0.1-random	0.2-best	0.2-random	Mixture	UMDA_comb	UMDA_unif
UMDA							
*f*1	90.08	**85.00**	96.08	87.92	129.08	244.75	170.11
*f*2	113.61	**109.08**	119.75	117.97	142.47	270.72	209.50
*f*3	72.44	**69.11**	98.78	94.53	136.36	248.11	189.08
*f*4	129.17	129.28	126.47	122.39	132.89	**115.47**	133.92
*f*5	88.83	**83.50**	96.92	94.07	125.22	283.03	175.12
*f*6	82.79	**69.44**	97.54	94.57	142.44	228.56	173.40
*f*7	93.76	99.60	**96.35**	97.39	153.67	184.68	160.06
*f*8	124.36	186.61	132.72	154.47	242.08	**27.08**	235.28
*f*9	**104.97**	108.97	111.22	111.25	166.64	189.78	131.25
*f*10	**92.31**	96.92	103.14	106.25	152.53	189.94	144.42
*f*11	101.14	**104.58**	104.17	108.69	159.86	180.97	126.08
EMNA							
*f*1	317.94	337.78	314.17	324.78	400.31		
*f*2	281.39	290.03	277.39	279.56	386.53		
*f*3	316.92	334.25	311.67	322.61	404.14		
*f*4	313.14	347.14	309.47	330.75	407.92		
*f*5	307.64	328.78	303.86	315.92	395.11		
*f*6	319.06	343.25	314.81	329.72	402.42		
*f*7	317.58	351.17	319.14	337.61	387.00		
*f*8	262.44	300.81	252.31	275.97	403.86		
*f*9	307.33	323.42	312.56	321.94	408.67		
*f*10	316.92	344.58	313.56	330.25	407.19		
*f*11	314.86	347.33	314.08	331.86	404.36		

Table 10
Average error of the best sequential EDA configurations with and without elitism on the 50 dimensional functions.

Function	Elitism	NoElitism	Elitism vs. NoElitism *p*-values	NoElitism vs. Elitism *p*-values	
*f*1	**0.00E+00**[a]	1.87E−02		1.87E−02	9.90E−01
*f*2	**4.14E+01**[a]	5.29E+01	7.79E−15	1.00E+00	
*f*3	**7.79E+03**[a]	3.67E+05	3-56E−11	1.00E+00	
*f*4	3.42E+02	**2.52E+01**[a]	1.00E+00	6.93E−10	
*f*5	**0.00E+00** [a]	2.69E−02	2.06E−02	9.81E−01	
*f*6	7.79E−07	**3.35E−07**[a]	1.00E+00	6.95E−10	
*f*7	0.00E+00	0.00E+00	Equal results		
*f*8	1.01E+03	**3.82E+02**[a]	1.58E−14	1.00E+00	
*f*9	**1.34E−05**[a]	4.52E−05	7.91E−15	1.00E+00	
*f*10	0.00E+00	0.00E+00	Equal results		
*f*11	**1.20E−05**[a]	4.08E−05	7.07E−10	1.00E+00	

[a] Means that there are statistical differences, according to the Wilcoxon test, with significance level $\alpha = 0.05$.

comparison. Furthermore, this configuration obtains significant results against the IPOP-CMA-ES algorithm on two of the three statistical procedures.

4.4. Characterization of the configurations

Finally, an unsupervised learning algorithm was applied to all the configurations to extract the groups of configurations that have a similar performance. For each configuration, a vector of all the ranks among all the functions on all the dimensions was collected and used as a representative value for the configurations. Therefore, two configurations with similar ranks among all the functions would be considered similar configurations. Then, a density-based clustering algorithm that has obtained successful results in the literature, the DBScan algorithm [14], was applied to the ranks vector of each configuration. In order to obtain an explanation of the results, the C4.5 machine learning algorithm was applied to the configurations to describe the assigned clusters, generating the induction tree depicted in Fig. 20, which represents the relevant characteristics that belong to each cluster. For determining the quality of each cluster, the average rank of the configurations belonging to the same cluster was computed, obtaining the values: 124.21, 201.97, 329.96, 329.15, 349.52 for clusters 1 to 5, respectively.

These results have confirmed the conclusions from the previous studies, being the type of algorithm the parameter that most differentiates the configurations. As previously shown, the performance of the $EMNA_{global}$ configurations is con-

Table 11
Average error of the best configurations.

Function Params	UMDA_inds 512 inds 8 islands period: 20 top: ring rate: 0.1 selec: best	UMDA_models 1024 inds 16 islands period: 20 top: a2a comb.: mixture	EMNA_inds 512 inds 16 islands period: 20 top: ring rate:0.2 selec: best	EMNA_models 2048 inds 8 islands period: 10 top: a2a comb.: mixture	UMDA_seq 512 inds 512 inds	EMNA_seq 2048 inds 2048 inds	IPOP-CMA-ES
50 dimensions							
1	4.4313e−12	1.0634e−11	4.9951e+03	7.1143e+04	2.0824e−11	6.8696e+04	**0.0000E+00**
2	3.1938e+01	3.4822e+01	3.0421e+01	4.7805e+01	4.2742e+01	4.1830e+01	**2.7500E−11**
3	1.3504e+02	1.1059e+02	6.8602e+07	5.7479e+09	4.3232e+04	4.0348e+09	**7.9700E−01**
4	**7.7246e+01**	3.2294e+02	3.4539e+02	4.3426e+02	3.4151e+02	3.9933e+02	1.0500E+02
5	**5.4197e−12**	1.2826e−11	9.5357e+00	5.6903e+02	2.5879e−11	5.4317e+02	2.9600E−04
6	**3.7939e−07**	5.4462e−07	1.2461e+01	1.5550e+01	7.5503e−07	1.4613e+01	2.0900E+01
7	**0.0000e+00**	5.1742e−08	2.8285e+01	9.7157e−01	**0.0000e+00**	1.0063e−03	1.0400E−10
8	7.0627e+02	2.2470e+03	4.4102e+03	1.8413e+00	1.0612e+03	2.1343e−07	**0.0000E+00**
9	**5.8894e−06**	2.4904e−01	2.0750e+02	2.3655e+01	1.6201e−05	3.7887e+00	9.6600E+00
10	**0.0000e+00**	**0.0000e+00**	1.9100e+02	1.6309e+00	**0.0000e+00**	1.2013e−06	6.3500E+00
11	**5.4996e−06**	2.4245e−01	1.9938e+02	2.1993e+01	1.1746e−05	3.7286e+00	2.5700E+01
100 dimensions							
1	2.7014e−11	3.3498e−11	1.2774e+04	1.9579e+05	8.2882e−11	1.8760e+05	**0.0000E+00**
2	4.5335e+01	4.7167e+01	4.5463e+01	6.4141e+01	5.4462e+01	5.7281e+01	**1.5100E−10**
3	3.7374e+02	3.5143e+02	2.9343e+08	2.0560e+10	2.4425e+05	1.4742e+10	**3.8800E+00**
4	**1.6540e+02**	1.6845e+02	8.7609e+02	1.0834e+03	8.0014e+02	9.9166e+02	2.5000E+02
5	**1.5235e−11**	2.1413e−11	2.3963e+01	1.5224e+03	4.9912e−11	1.4321e+03	1.5800E−03
6	**5.8275e−07**	6.5915e−07	1.3217e+01	1.8225e+01	9.8306e−07	1.7581e+01	2.1200E+01
7	**0.0000e+00**	1.7800e−10	7.0014e+01	6.8451e+00	**0.0000e+00**	6.3445e−01	2.4300E−05
8	4.4043e+03	1.7445e+04	1.1765e+04	4.9132e+01	2.3255e+04	5.0400e−01	**0.0000E+00**
9	2.4198e−08	1.6342e−01	4.1550e+02	9.5136e+01	**8.8410e−09**	2.9663e+01	5.4000E+01
10	**0.0000e+00**	**0.0000e+00**	4.9928e+02	2.8885e+01	**0.0000e+00**	6.6738e−01	1.7300E+01
11	1.2051e−07	1.4185e−01	4.1484e+02	9.5033e+01	**7.3676e−09**	2.5700e+01	1.7200E+02
200 dimensions							
1	1.0243e−10	1.2379e−10	4.4513e+04	1.7268e+05	1.7257e−03	4.8020e+05	**0.0000E+00**
2	5.7000e+01	6.1222e+01	6.6971e+01	8.9225e+01	6.5054e+01	6.9707e+01	**1.1600E−09**
3	5.5483e+02	7.3906e+02	2.7326e+09	3.2678e+10	1.6448e+06	7.8684e+10	**8.9100E+01**
4	3.6909e+02	**2.8513e+02**	2.1671e+03	3.4675e+03	1.7958e+03	2.5607e+03	6.4800E+02
5	3.4918e−11	4.5310e−11	8.8792e+01	3.8037e+02	8.4807e−11	3.6309e+03	**0.0000E+00**
6	**8.0757e−07**	8.6880e−07	1.5303e+01	1.9868e+01	1.2352e−06	1.9535e+01	2.1400E+01
7	**0.0000e+00**	2.4687e−13	1.8994e+02	1.0000e+11	**0.0000e+00**	1.5461e+01	6.3200E−02
8	1.5540e+04	1.0114e+05	5.2883e+04	2.5319e+05	1.4260e+05	1.1979e+02	**0.0000E+00**
9	2.1486e−01	4.1210e−01	9.2002e+02	1.4805e+03	**1.2772e−13**	2.1215e+02	2.4900E+02
10	**0.0000e+00**	**0.0000e+00**	1.5617e+03	1.5511e+04	**0.0000e+00**	6.1790e+01	4.3100E+01
11	2.5025e−01	3.2275e−01	9.1155e+02	1.4782e+03	**1.2612e−13**	2.1663e+02	8.0100E+02

Table 12
Statistical validation (UMDA_inds is the control algorithm).

UMDA_inds vs.	*z*-value	*p*-value	Holm *p*-value	Hochberg *p*-value
UMDA_model	1.99E+00	4.61E−02	4.61E−02[a]	4.61E−02[a]
EMNA_inds	6.55E+00	5.65E−11	2.82E−10[a]	2.82E−10[a]
EMNA_model	8.15E+00	4.44E−16	2.66E−15[a]	2.66E−15[a]
UMDA_seq	2.45E+00	1.43E−02	3.10E−02[a]	2.86E−02[a]
EMNA_seq	5.81E+00	6.17E−09	2.47E−08[a]	2.47E−08[a]
IPOP-CMA-ES	2.56E+00	1.03E−02	3.10E−02[a]	2.86E−02[a]

[a] Means that there are statistical differences with significance level $\alpha = 0.05$.

siderably lower than the equivalent $UMDA_g$ configurations having all the $UMDA_g$ clusters (1 and 2) a superior cluster average rank that the $EMNA_{global}$ clusters (3, 4 and 5). For the $UMDA_g$ configurations, the second most important attribute is the information exchange method, being all the configurations based on sending individuals grouped in the same cluster plus some of the configurations based on sending models (the ones with the best results). For the $EMNA_{global}$ configurations, it seems that the population size per island determines better the performance of the configuration than the information exchange method.

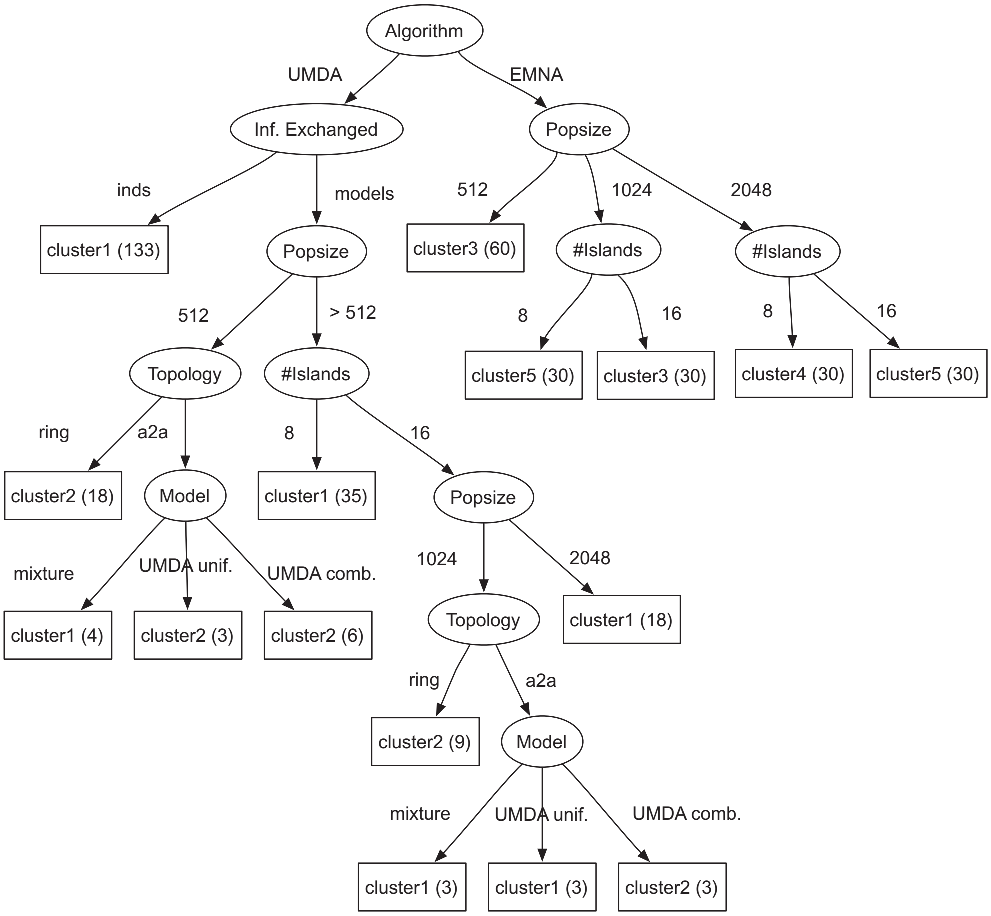

Fig. 20. C4.5 Induction tree from the DBScan cluster. Each leaf node contains the cluster name and, in parentheses, the number of instances classified by the branch. The average ranks of the configurations included in each cluster are: cluster1: 124.21, cluster2: 201.97, cluster3: 329.96, cluster4: 329.15 and cluster5: 349.52.

5. Conclusions

This paper presents an extensive comparison of several configurations of dEDAs and sequential EDAs over a standard benchmark of continuous functions in both 50, 100 and 200 dimensions. Several analyses from different points of view have been carried out and non-parametric tests have been applied for contrasting the achieved results. From these studies, it has been observed than the learning model of the algorithms has been the most decisive factor. It seems that, for the proposed benchmark, the $UMDA_g$ has obtained the best results. However, it must be taken into account that the $EMNA_{global}$ algorithm needs a considerable amount of individuals to induce the dependencies among the variables and, with the benchmark constraint on the number of evaluations, this has implied a considerable reduction on the number of iterations which has heavily penalized the final results. Therefore, a future work would be to extend this analysis with a higher number of evaluations and also with the more complex *EGNA* models which also require larger population sizes.

A special attention has been paid to determine which method for exchanging information between dEDAs, the migration of individuals or the migration of probabilistic models, is the best approach for a researcher who would like to apply the $UMDA_g$ or $EMNA_{global}$ dEDAs in a continuous domain. From this perspective, the results from this study clearly express that, for most of the functions, the exchange of individuals obtains significantly better results than the alternative approach of sending models, although the remaining parameters must be properly tuned to obtain the best results. However, it was also discovered that, for the functions where the absence of elitism improves the results, the exchange of individuals (which implies a degree of elitism) obtains worse results.

From the remaining parameters of the distributed configurations some interesting relationships have arisen such as the interaction between the topology and the criterion for selecting the immigrants, the relationship between the topology and the population size, and between the population size, migration period and topology with the selected method for conducting the combination of the models.

The question of whether the dEDAs configurations obtain better results than their equivalent sequential versions has been also addressed: the study shows that the best dEDAs configurations outperform the best results of the sequential counterparts.

Finally, taking advantage of all the configurations analyzed through this experiment, a characterization of the behavior of the configurations has also been presented. The clustering analysis of all these configurations shows that the type of algorithm (*UMDA*$_g$ and *EMNA*$_{global}$), the information exchange method for the *UMDA*$_g$ configurations and the population size per island for the *EMNA*$_{global}$ configurations are the most determinant characteristics for characterizing the behavior of the configurations.

Acknowledgments

This work was financed by the Spanish Ministry of Science TIN2010-21289-C02-02 and supported by the Cajal Blue Brain Project. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid (CeSViMa) and the Spanish Supercomputing Network. The authors would also like to thank Nelis Franken for his help with his Fluxviz tool.

References

[1] Chang W. Ahn, David E. Goldberg, R.S. Ramakrishna, Multiple-deme parallel Estimation of Distribution Algorithms: basic framework and application, in: Parallel Processing and Applied Mathematics, 2004, pp. 544–551.

[2] Enrique Alba, Parallel evolutionary algorithms can achieve super-linear performance, Information Processing Letters 82 (1) (2002) 7–13.

[3] Enrique Alba, José M. Troya, Influence of the migration policy in parallel distributed gas with structured and panmictic populations, Applied Intelligence 12 (3) (2000) 163–181.

[4] Enrique Alba, José M. Troya, Improving flexibility and efficiency by adding parallelism to genetic algorithms, Statistics and Computing 12 (2) (2002) 91–114.

[5] S. Baluja, Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, Technical report, Carnegie Mellon University, 1994.

[6] S. Baluja, S. Davies, Combining multiple optimization runs with optimal dependency trees, Technical report, Justsystem Pittsburgh Research Center & Carnegie Mellon University, 1997.

[7] J.S. De Bonet, C.L. Isbell, P. Viola, Structure driven image database retrieval, Advances in Neural Information Processing, vol. 9, MIT Press, 1997, p. 424.

[8] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers., 2001.

[9] J.P. Cohoon, S.U. Hegde, W.N. Martin, D.S. Richards, Punctuated equilibria: a parallel genetic algorithm, in: Proceedings of the 2nd International Conference on Genetic Algorithms, July 1987, pp. 148–154.

[10] L. delaOssa, J.A. Gámez, J.M. Puerta, Initial approaches to the application of islands-based parallel EDAs in continuous domains, in: Proceedings of 2005 International Conference on Parallel Processing Workshops, 2005, pp. 580–587.

[11] L. delaOssa, J.A. Gámez, J.M. Puerta, Initial approaches to the application of islands-based parallel EDAs in continuous domains, Journal of Parallel and Distributed Computing 66 (8) (2006) 991–1001.

[12] Luis delaOssa, José A. Gámez, José M. Puerta, Migration of probability models instead of individuals: an alternative when applying the island model to EDAs, in: PPSN VIII, 2004, pp. 242–252.

[13] W. Dong, X. Yao, Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms, Information Sciences 178 (15) (2008) 3000–3023.

[14] Martin Ester, Hans-Peter Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, vol. 96, 1996, pp. 226–231.

[15] N. Franken, Visual exploration of algorithm parameter space, in: Proceedings of the Eleventh Congress on Evolutionary Computation, 2009, pp. 389–398.

[16] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (10) (2010) 2044–2064.

[17] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 special session on real parameter optimization, Journal of Heuristics 15 (6) (2009) 617–644.

[18] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Transactions on Evolutionary Computation 3 (3) (1999) 287–297.

[19] J. Jaros, J. Schwarz, Parallel BMDA with probability model migration, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2007, pp. 1059–1066.

[20] J. Jaros, J. Schwarz, Parallel BMDA with an aggregation of probability models, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2009, pp. 1683–1690.

[21] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization by learning and simulation of Bayesian and Gaussian networks, Technical report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.

[22] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Combinatorial optimization by learning and simulation of bayesian networks, in: Proceedings of the 16th Conference on Uncertainty in Articial Intelligence, UAI 2000, 2000, pp. 343–352.

[23] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, in: Proceedings of the Workshop in Optimization by Building and using Probabilistic Models. A Workshop within the Genetic and Evolutionary Computation Conference, 2000, pp. 201–204.

[24] P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Springer, 2002.

[25] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation, Kluwer Academic Publisher, 2002.

[26] L-V. Lozada-Chang, R. Santana, Univariate marginal distribution algorithm dynamics for a class of parametric functions with unitation constraints, Information Sciences 181 (11) (2011) 2340–2355.

[27] Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, Endika Bengoetxea, Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms (Studies in Fuzziness and Soft Computing), Springer-Verlag, New York, Inc., 2006.

[28] Julio Madera, Enrique Alba, Alberto Ochoa, A parallel island model for estimation of distribution algorithms, in: Towards a New Evolutionary Computation, Studies in Fuzziness and Soft Computing, vol. 192, Springer, Berlin/Heidelberg, 2006, pp. 159–186.

[29] A. Mendiburu, J.A. Lozano, J. Miguel-Alonso, Parallel implementation of EDAs based on probabilistic graphical models, IEEE Transactions on Evolutionary Computation 9 (4) (2005) 406–423.

[30] Santiago Muelas, José M. Peña, Víctor Robles, Antonio LaTorre, Pedro de Miguel, Machine learning to analyze migration parameters in parallel genetic algorithms, in: Innovations in Hybrid Intelligent Systems, Advances in Soft Computing, vol. 44, 2007, pp. 199–206.

[31] H. Mühlenbein, The equation for response to selection and its use for prediction, Evolutionary Computation 5 (3) (1997) 303–346.

[32] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions i. binary parameters, in: Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature – PPSN IV, 1996, pp. 178–187.

[33] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, BOA: the bayesian optimization algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, 1999, pp. 525–532.

[34] M. Pelikan, H. Mühlenbein, The bivariate marginal distribution algorithm, in: R. Roy, T. Furuhashi, P.K. Chawdhry (Eds.), Proceedings of Advances in Soft Computing – Engineering Design and Manufacturing, 1999, pp. 521–535.

[35] Martin Pelikan, David E. Goldberg, Fernando Lobo, A survey of optimization by building and using probabilistic models, Computational Optimization and Applications 21 (1) (2002) 5–20.

[36] Martin Pelikan, Kumara Sastry, Erick Cantú-Paz, Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications (Studies in Computational Intelligence), Springer-Verlag, New York, Inc., 2006.

[37] Chrisila Petty, Michael Leuze, A theoretical investigation of a parallel genetic algorithm, in: Proceedings of the 3rd International Conference on Genetic algorithms, 1989, pp. 398–405.

[38] Kumara Sastry, David E. Goldberg, Xavier Llorà, Towards billion-bit optimization via a parallel estimation of distribution algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2007, pp. 577–584.

[39] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the special session on real parameter optimization, Technical report, Nanyang Technological University, 2005.

[40] G. Taguchi, S. Chowdhury, Y. Wu, Taguchi's Quality Engineering Handbook, John Wiley, 2005.

[41] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.P. Chen, C.M. Chen, Z. Yang, Benchmark functions for the CEC 2008 special session and competition on large scale global optimization, Technical report, Nature Inspired Computation and Applications Laboratory, USTC, 2007.

[42] D. Whitley, S. Rana, R.B. Heckendorn, The island model genetic algorithm: on separability, population size and convergence, Journal of Computing and Information Technology 7 (1999) 33–47.

[43] A.A. Zhigljavsky, Theory of Global Random Search, Kluwer Academic Publishers, 1991.