Avoiding premature convergence in Estimation of Distribution Algorithms

Luis delaOssa, José A. Gámez, Juan L. Mateo and José M. Puerta

Abstract— **This work studies the problem of premature convergence due to the lack of diversity in Estimation of Distributions Algorithms. This problem is quite important for these kind of algorithms since, even when using very complex probabilistic models, they can not solve certain optimization problems such as some deceptive, hierarchical or multimodal ones.**

There are several works in literature which propose different techniques to deal with premature convergence. In most cases, they arise as an adaptation of the techniques used with genetic algorithms, and use randomness to generate individuals.

In our work, we study a new scheme which tries to preserve the population diversity. Instead of generating individuals randomly, it uses the information contained in the probability distribution learned from the population. In particular, a new probability distribution is obtained as a variation of the learned one so as to generate individuals with less probability to appear on the evolutionary process.

This proposal has been validated experimentally with success with a set of different test functions.

Index Terms— **Estimation of Distribution Algorithms, Diversity, Premature convergence.**

I. Introduction

Equilibrium between information exploitation and exploration in the solution space is a key point for the performance of metaheuristic techniques.

In case of genetic algorithms, and evolutionary algorithms in general, information exploitation is produced due to selection pressure, which gives individuals with better fitness a bigger probability of surviving and being selected to generate subsequent generations.

On the other hand, and despite the fact that mutation operator was originally conceived as the only way to provide exploration capacity, this property also depends on the selection and replacement operators [1]. Thus, too high selection pressure leads the algorithm towards the search among a relatively reduced subset of solutions. This phenomena is known as premature convergence, and entails a risk of stagnation of the process in local optima. Because of that, and due to the fact that mutation effects are discused by several authors, different proposals concerning to these operators which try to preserve the diversity of solutions in the population have arisen [2].

Estimation of Distribution Algorithms (EDAs) present, in that sense, several differences with respect to Genetic Algorithms (GAs). EDAs, which learn a probability model from the best solutions and use it to generate new populations, improve the performance of GAs since they are able to detect and exploit dependences among variables. However, the fact that all individuals in a population are sampled from the same model increases the lack of diversity in some cases. In fact, there are many algorithms of this family which use specific replacement operators to deal with this problem [3].

This work presents an alternative to improve the exploration capability of EDAs. The fact that the probability model contains information about the individuals in a population makes possible to identify which zones in the search space will be explored more intensely, but also those which will not. Such information can be taken advantage of so that the algorithm generates diverse solutions and does not stagnate in a local optima.

This article is divided into 5 sections besides this introduction. Section 2 describes the state of the art in preservation of diversity on EDAs. Next, Section 3 explains the motivation of this work, and Section 4 describes with detail the proposed method. Afterwards, in Section 5, an experimental study and an analysis of the results are carried out. Last, Section 6 presents our conclusions and proposals for future work.

II. Diversity and EDAs

As commented above, preservation of diversity is a key point to avoid premature convergence and stagnation in local optima of evolutionary algorithms. This property is, moreover, essential for the success of this kind of algorithms when solving more complex problems, such as the multimodal, hierarchical, dynamic, or multiobjective ones.

The main techniques for preservation of diversity try to maintain several subpopulations or *niches* [4], each one containing solutions belonging to a different region of the search space, in the evolutionary process.

Traditional niche-based methods modify the fitness function (*fitness-sharing*) [5], or the selection or replacement mechanisms. In the first case, a penalty factor proportional to the size of the niche which the individual belongs to is used. In the second approach, however, the selection or replacement processes are modified so that they consider diversity. In this case, the most popular method, *crowding*

Authors are in the Department of Computing Systems, Intelligent Systems and Data Mining Lab, University of Castilla-La Mancha, Albacete 02071, Spain (email {ldelaossa,jgamez,juanlmc,jpuerta}@dsi.uclm.es)

978-1-4244-2959-2/09/$25.00 ©

[6], replaces with a generated individual that one in a previously selected subset which is nearest to it in the solution space.

Niche-based techniques allow to identify multiple optima (potentially as many as niches) and, in the case of EDAs, they open other possibilities, such as the obtaining of better solutions by means of the suitable treatment of the probability models which represent each niche.

One of the first attempts to preserve diversity on EDAs, based on the methods mentioned above, was the *hierarchical Bayesian Optimization Algorithm* (hBOA) [7], which uses a technique called *Restricted Tournament Replacement* (RTR) [8]. The main difference between this technique and the classical method of crowding is that RTR is a replacement technique. Thus, for each new individual sampled from the current probability model, a subset of individuals from the current population is selected. Then, the fitness of such individual is compared with the fitness of the one which is more similar to it, preserving for the next population the one with higher fitness. It is worth pointing out that the size of the subsets used for RTR is set up depending on the population size and the complexity of the problem being solved.

RTR is probably the most extended method for preservation of diversity on EDAs. However, and due to nature of this kind of algorithms, there have been other attempts which are not based on individuals, but they try to reflect diversity on the probabilistic models learned from the population at each generation.

The main idea which underlies these works is using probabilistic models, in general, to induce or discover partitions of the individuals in the population. Depending on the way this is carried out, different EDAs arise. Thus, models based on fitness classification can be found [9]. In such models, individuals are labeled according to their fitness to subsequently learn a Bayesian classifier (Naive Bayes, TAN, etc.). Once the classifier has been learned, a subset of individuals is sampled for each one of the classes. Then, the best ones are selected as part of the next population.

There are other approaches based on probabilistic clustering [10]. In them, a non supervised clustering of the population is carried out and a probabilistic model is learned to generate the next population. These models are based on Bayesian classifiers with a hidden class/cluster variable, and use the EM algorithm to make the estimation.

In other proposals, the clustering is carried out by non probabilistic techniques based on neighborhood, such as the k-means algorithm [11], but the underlying idea of preserving diversity is the same.

Other approach to preserve diversity on EDAs, the one our work is based on, introduces mutation [12]. In this work, the complementary probability for some problem variables is used, according with a probability which is set up as parameter, to sample some individuals. The idea of mutation is also present in [13] where, after updating the model with the selection of the best individuals of the population, the parent set, the *guided mutation* is applied to this set in the way that only some bits will be modified according with the learned probabilistic model and the other will keep the previous value. The rate of modified or mutated bits is given the the mutation probability parameter.

III. Motivation

As it has been mentioned above, there has been some attempts to preserve diversity on EDAs. Besides those inherited from GAs, there are other which divide the populations into groups and learn different probabilistic models from them instead of learning one global model.

Anycase, these methods only deal with individuals which fit in the probabilistic models which have already been learned, since they have been generated from these models. Therefore, once the algorithm has been lead to a subspace with a local optima, it is very difficult for it to escape. The only option to do that is probably the randomness in the sampling process, since sometimes some individuals that allow the algorithm to escape from that situation could be generated. However, as the evolution process advances, the probability of that to happen decreases.

In Figure 1, a graphical representation of the commented problem is shown. The cloud represents the whole search space of the problem, and the dots the initial population of the algorithm. Only a percentage of these individuals is taken into account to learn the model and, specially for complex problems as, for instance the deceptive, this selection will lead the evolutive process towards local optima.

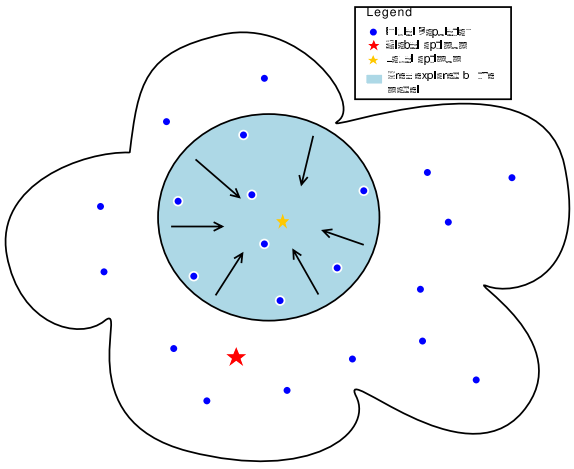

Fig. 1. Representation of the problem of premature convergence for EDAs. Arrows show the model pressure towards a certain point in the search space. The model will only explore the shaded region.

Because of this reason, we consider that it could be beneficial for EDAs performance an scheme for preserving diversity which introduces individuals with low probability of being sampled from the probabilistic model.

The aim of the method proposed in [12] is to achieve this goal by introducing a mutation operator in the probability distribution. In particular, the probability distribution of a certain variable is changed by its complementary. As this proposal entails a random generation of the individuals, the probability of mutation must be low. Otherwise, the convergence would be affected in a critical way. In the other hand, mutating a small set probabilities could have no consequences in the evolution process since, in many cases, there are dependences among variables and, if the variables which dependent on the mutated one preserve their values and probabilities, the induced change might not lead to better individuals and would be nulled in one or two generations. In other words, if the mutated variable breaks a building block the global fitness would be worse and the change produced by the mutation ignored. Therefore, no diversity would be induced.

A different proposal consists of using the information which is contained in the probabilistic model to generate individuals which introduce diversity. In particular, instead of mutating the probability of the variables randomly, and generate individuals similar to those already contained in the population, it could be worth generating individuals far from the existing ones. This way, instantiations of group of variables which otherwise would have a quite low probability to be generated, would appear in the population, but would not slow down the convergence process as they would be discarded if the solutions which contain them had not a high fitness. However, if these solutions were preserved the probabilistic model would gather such changes. This idea is shown in Figure 2.

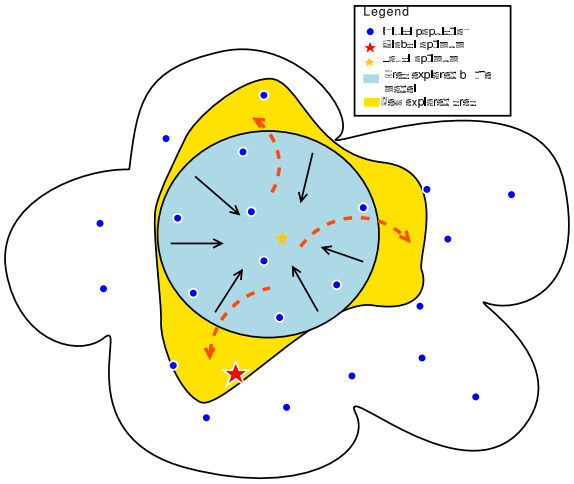

Fig. 2. Representation of the proposal of using information gathered in the model in order to improve EDAs robustness. Non-solid arrows indicate escape ways from a premature convergence.

The question is now how to specify the way such individuals are generated and determining which is the right proportion of such individuals. Different proposals can arise depending on the way these questions are answered, however, this work only explains one of them, which can serve as a basis for the developing of other proposals in a near future. Next section explains the solution proposed.

IV. Sampling rebel individuals

As we have mentioned above, the aim of this work is maintaining diversity in the population during all iterations in order to prevent the algorithm from stagnating in local optima. In particular, we have focused on binary combinatorial optimization problems. Thus, some details depend on the individuals codification and, therefore, on the features of the probability distributions used.

Next, we will describe with detail our proposal.

A. Counter evolution specification

There are many options to generate individuals different from those currently present in a given population. The proposed mechanisms could deal with the probabilistic model, both with its structural and parametrical part, with the distance among sampled individuals or their fitness.

We have considered a simple approach which only uses the parametric part of the model. In particular, we propose that "rebel" individuals are sampled from a distribution which is the inverse of the learned one. This option implies that such individuals are the complementary of those which would have been obtained from the current probability distribution, i. e., the individuals would be obtained by inverting the value of each variable in a binary model. Although it could seem a quite meaningless solution, its founding is not so. In fact, in problems with continuous variables or even discrete the idea could be studied deeply. However, the analysis of such situation is left for future work.

The idea of using the inverse probability is not new. In previous works as [12] it has been used to determine the new probability of the mutated bits. The expression which defines the inverse probability for finite variables is:

$$P(\overline{x_i}) = \frac{1 - p(x_i)}{|X_i| - 1},$$

where $|X_i|$ is the number of states of the variable.

The main difference of the proposed method is that, in this cases, probabilities are inverted for all variables, but only a limited number of individuals will be generated.

B. Proportion of rebel individuals

The other question which arises is how many individuals should be sampled from the current probability distribution and how many from the inverted one. As the sum of both quantities must be equal to the population size determined, it becomes necessary to specify the percentage of the population size which will be sampled from each one of the models. In particular, we will refer to the percentage of individuals sampled from the inverted model.

A huge value for this parameter, higher than 50%, would deteriorate the convergence, since the information contained in the best individuals would not be exploited properly. Intuitively, it seems reasonable considering values under 30% for generating spoiled individuals. In particular, we have

considered a value of 15% basing in some preliminary experiments, since this is the value that best behaves, preserving convergence properties and leading the algorithm towards solutions whereas preserving an acceptable level of diversity which allows such algorithm to escape from local optima. With lower values the benefits of introducing rebel individuals is hardly seen, whereas higher values, as mentioned affect the convergence properties of the algorithms.

□

Considering what has been explained so far, the mechanism of diversity preservation is stated as the pseudo-code shown in Figure 3.

1 $l \leftarrow 0$ /* Generation counter */
2 $D_l \leftarrow$ Generate M individuals randomly
3 $pct \leftarrow 0.15$ /* Fix the percentage of rebel individuals */
4 **repeat**
5 $\quad l \leftarrow l + 1$
6 $\quad D_{l-1}^{Se} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to the selection method
7 $\quad p_l(\mathbf{x}) = p(\mathbf{x}|D_{l-1}^{Se}) \leftarrow$ Estimate the probability model from the selected individuals
8 $\quad D_{ok} \leftarrow$ Sample $M * (1 - pct)$ individuals from $p_l(\mathbf{x})$
9 $\quad D_r \leftarrow$ Sample $M * pct$ individuals from $p_l(\overline{\mathbf{x}})$
10 $\quad D_l \leftarrow D_{ok} \cup D_r$
11 **until** *stop condition*

Fig. 3. Pseudo-code of the proposed mechanism

The difference between this algorithm and the standard EDA lies on lines 8, 9, and 10, where individuals are first sampled according to the probability distribution learned in the former step. Then, a certain amount of rebel individuals is sampled. Last, the population used to learn the probabilistic model from which next generation will be sampled is the union from the two sets of individuals.

V. Experimental evaluation

In order to analyze the performance of our proposal we show some experiments. Mainly, we compare our algorithm against the classic EBNA [14] and hBOA [7]. In all cases we have used the BIC score metric [15] and a greedy algorithm (hill climbing) which considers as local operations to create a neighbors link addition and removal in the structural learning of the model. Our proposal, which will be labelled as EBNA$_r$, has been generated according the pseudo-code in Figure 3 and employing as base model the multivariate EBNA. For each configuration we have performed 50 runs independent among them, but we have ensured that each one of the 50 runs has the same initial population for all the algorithms in order to avoid bias in the results by effect of the initial population.

We have selected 5 functions with different characteristics: HIFF [16], EqualProducts [17], SixPeaks [17], Knapsack [18] and HTRAP [7]. In all algorithms we have used the same configuration, which has been taken from [12] in order to be able to fairly compare with the algorithm presented there where the function HIFF is also used. In that work authors point out that the best result obtained is with a configuration with 3 classes, $|K| = 3$, and selecting the individuals represented with the best fitness, $|C| = 1$. According with that we pick one third of the population with best individuals to learn the model. We repeat the experiments for two population sizes, 500 and 1000 individuals. We set 200 generations at maximum as stop condition with both population sizes. Algorithm also ends when the optimum is reached. All experiments have been developed using the library for metaheuristic algorithms LiO [19].

Next we show a brief description of the functions used.

A. Functions

- HIFF

The HIFF function (hierarchical if and only if) was defined in [16]. For this function, the input string must be an integer power of 2, and the power l is the number of levels. The structure of the input string can be seen as a balanced binary tree where l is its depth. Each leaf contributes to the fitness by 1 and the tree have to be processed up to the root in such a way that every parent node x contributes to the fitness by $2^{height(x)}$, where $height(x)$ is the distance of x to the leaves, but that node only contribute if both children are interpreted either 0 or 1, otherwise the contribution is 0. A child node is interpreted by its parent as 0 if both children of that variable are 0, is interpreted as 1 if both children are 1, other wise is interpreted as null symbol.

This problem can be defined in a functional way as follows:

$$h_{iff}(\mathbf{x}) = \begin{cases} 1 & \text{if } p = 0 \\ 1 & \text{if } h_{iff}(L) = 1, h_{iff}(R) = 1 \text{ and } L = R \\ 0 & otherwise \end{cases}$$

$$HIFF(\mathbf{x}) = HIFF(L) + HIFF(R) + \begin{cases} length(\mathbf{x}) & \text{if } h_iff(\mathbf{x}) = 1 \\ 0 & otherwise \end{cases}$$

This function has two global optimum which are the strings in which all bits have a zero or a one value respectively. The local optima closer to those global optima are the two strings in which the right half is the complementary of the other half and all bits in each half have the same value.

- EqualProducts

This function was presented in [17], and for it we need a set of n random real numbers $\{a_1, a_2, \ldots, a_n\}$ within the range $[0, k]$. The goal of this problem is to make two disjoints subsets with these numbers in such a way that the difference between the product of all the numbers in the same subset is as small as possible. This function can be written as follows:

$$F_{EqualProducts}(\mathbf{x}) = \left| \prod_{i=1}^{n} h(x_i, a_i) - \prod_{i=1}^{n} h(1 - x_i, a_i) \right|$$

where the function h is defined as:

$$h(x, a) = \begin{cases} 1 & \text{if } x = 0 \\ a & \text{if } x = 1 \end{cases}$$

Since the set of numbers is random, the optimum value for this problem cannot be known. Nonetheless that value should as near to zero as possible.

- SixPeaks

This problem were defined in [17] too and can be written in a functional way as follows:

$$F_{SixPeaks}(\mathbf{x}, t) = \max\{tail(0, \mathbf{x}), head(1, \mathbf{x}), tail(1, \mathbf{x}), head(0, \mathbf{x})\} + \mathcal{R}(\mathbf{x}, t)$$

where

$$tail(b, \mathbf{x}) = \text{number of } b\text{'s at the end of } \mathbf{x}$$

$$head(b, \mathbf{x}) = \text{number of } b\text{'s at the beginning of } \mathbf{x}$$

$$\mathcal{R}(\mathbf{x}, t) = \begin{cases} n & \text{if } (tail(0, \mathbf{x}) > t \text{ and} head(1, \mathbf{x}) > t) \text{ or} \\ & \quad (tail(1, \mathbf{x}) > t \text{ and } head(0, \mathbf{x}) > t) \\ 0 & \text{otherwise} \end{cases}$$

The name of this function is due to the fact that it has six maximums or peaks, 4 of them are global optimums and the other two are local optimums. The difficulty of this problem is that these two local optimums, which are represented by these strings $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$, are very easy to get. In the other hand, the global optimums are isolated and are these:

$$(\overbrace{0, 0, \ldots, 0}^{t+1}, 1, 1, \ldots, 1) \qquad (\overbrace{1, 1, \ldots, 1}^{t+1}, 0, 0, \ldots, 0)$$

$$(0, 0, \ldots, 0, \overbrace{1, 1, \ldots, 1}^{t+1}) \qquad (1, 1, \ldots, 1, \overbrace{0, 0, \ldots, 0}^{t+1})$$

- Knapsack

We also use the known knapsack problem, in which we have a bag with a limited capacity and the goal is to put into it as many objects as possible which give us the maximum benefit but without going beyond the bag limit. We have selected the software developed by David Pisinger [18] which generates instances of this problem. Specifically we consider a instance with 64 elements whose values for weight and utility are in the rage 0-100 and they are not correlated. The total capacity of the bag is 101. We use a penalizing factor, K, in the case a configuration exceed the maximum capacity which multiply its fitness:

$$K = \frac{\min(w_i)}{((\sum_{i=0}^{i=n} w_i) - capacity)},$$

where w_i is the weight of the i element in the bag.

- HTRAP

This function (Hierarchical TRAP) [7] is also hierarchical like HIFF. In this case, the structure is a balanced k-ary tree instead of a binary tree, where $k \geq 3$, so the input string is of size k^l. Every group of k children is interpreted in a similar way as in HIFF function: if they all are 0 the interpretation is 0, if all are 1 the interpretation is 1, and the null symbol otherwise. In the implementation used here the leaves do not contribute to the global fitness. The contribution to the fitness for every group is measure by a version of the TRAP function whose difference is that return values in the range $[0, 1]$:

$$trap(\mathbf{x}) = \begin{cases} f_{high} & \text{if } u = k, \\ f_{low} \cdot \dfrac{k - 1 - u}{k - 1} & otherwise \end{cases}$$

If in the input of this function there is a null symbol it returns 0 whatever were the other values. For all nodes x but the root the trap function is applied with $f_{high} = 1$ and $f_{low} = 1 + 0.1/l$, which biases the optimum to a string with k 0's. For the root node, $f_{high} = 1$ and $f_{low} = 0.9$, which biases the optimum to a string full of 1's. Thus the global optima is the string with all bits equal to 1, while the local optima is the string with all bits to 0.

To make to overall contribution at each level of the same magnitude, the contributions of traps on the $i-th$ level from the bottom up to the root are multiplied by k^i.

□

For all functions we have chosen a configuration in order to have a problem representation with strings of 64 bits. Therefore, for HIFF function we have set the number of levels to $l =$ and for HTRAP function the trap size is $k = 4$ and the number of levels $l = 3$.

B. Results

In this subsection we enumerate the results obtained in our experimentations and their analysis. For each function we show a table which contains, for each population size, the average the best fitness value reached for the 50 runs, the number of evaluations needed to get such fitness value, the total number of evaluations in the execution and the runs in which the optimum value was found. In addition, we have performed a statistical analysis to determine whether there exists significant difference between the three models compared regarding their capacity to reach the optimum fitness value. We use the paired t-test with the typical confidence level of 95%. With the symbol • we indicate the best results and with the symbol ◦ we indicate those results which are not significantly different with the best one according with the test.

We start with the results for the HIFF function which are shown in Table I. In both cases, population sizes 500 (a) and 1000 (b) individuals, the best models are hBOA and EBNA$_r$, however we can observe that EBNA$_r$ needs less evaluations

than hBOA to obtain the same results, i.e. 20% and 10% of reduction respectively.

TABLE I

RESULTS FOR THE FUNCTIONS HIFF WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	evals. to best	total evals.	succeed
EBNA	405.76	6970	57210	23
hBOA	•441.60	9694	19070	45
EBNA$_r$	○440.32	7672	17290	45

(a)

EBNA	440.96	12418	31840	45
hBOA	•448.00	13056	13660	50
EBNA$_r$	•448.00	11765	12400	50

(b)

In order to compare those three models with the one proposed in [12], since in that work authors do not provide the actual numbers but a graphical representation we have made the same charts. In Figures 4 and 6 we can see the charts taken from [12], while in Figures 5 and 7 we can see the charts regarding EBNA, hBOA and EBNA$_r$ models.

With a population size of 500 individuals we can see in Figure 4 that the best result is obtained with a sampling mutation probability of 0.05 but the convergence is reached at 20000 evaluations and is far from the optimum fitness. In Figure 5 we can see that EBNA gets a performance similar to EBCOA model without sampling mutation, nonetheless both hBOA and EBNA$_r$ get much better results closer to the optimum than EBCOA with 0.05 SM and the plateau state is reached with less evaluations, 14000 and 9000 evaluations respectively.

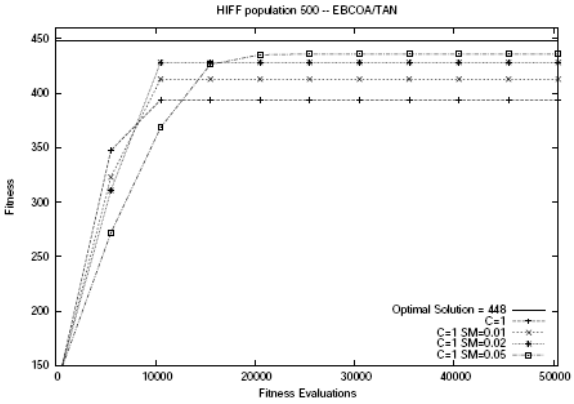

Fig. 4. The 64-bit HIFF problem. Comparison of best fitness using EBCOA/TANB with $|C| = 1$ and $|K| = 3$ classes without sampling mutation and with 0.01, 0.02 and 0.05 respectively. Population size is 500 individuals. Image taken directly from [12].

With a population size of 1000 individuals we can observe something similar as before. In this case, EBCOA with SM 0.02 and 0.05 reaches the line indicating the optimum value, Figure 6, what is also accomplished by hBOA and EBNA$_r$, Figure 7, but the later two model can do it in less than 20000 evaluation meanwhile the former ones need 20000 or more than 30000 evaluations respectively.

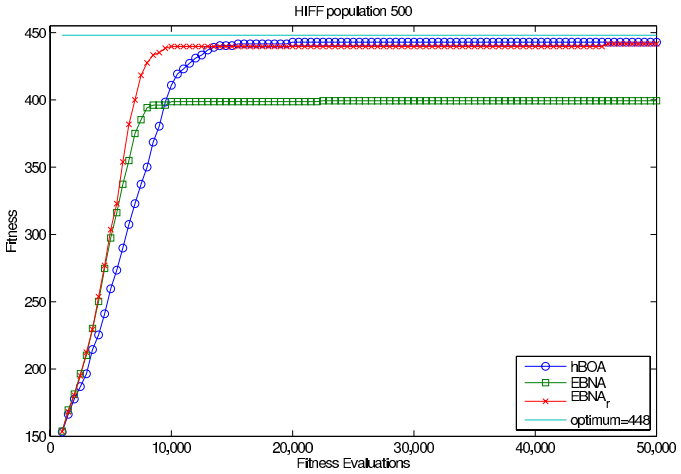

Fig. 5. The 64-bit HIFF problem. Comparison of best fitness using EBNA, hBOA and EBNA$_r$. Population size is 500 individuals.

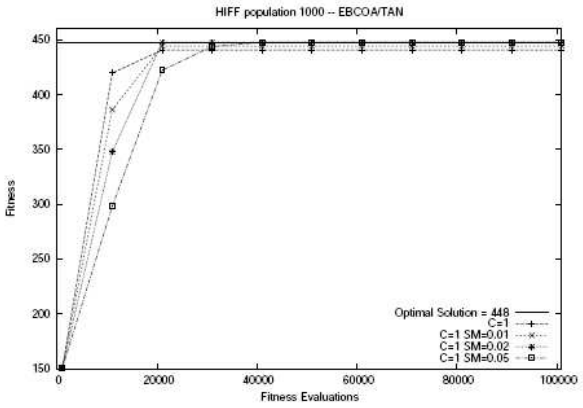

Fig. 6. The 64-bit HIFF problem. Comparison of best fitness using EBCOA/TANB with $|C| = 1$ and $|K| = 3$ classes without sampling mutation and with 0.01, 0.02 and 0.05 respectively. Population size is 1000 individuals. Image taken directly from [12].

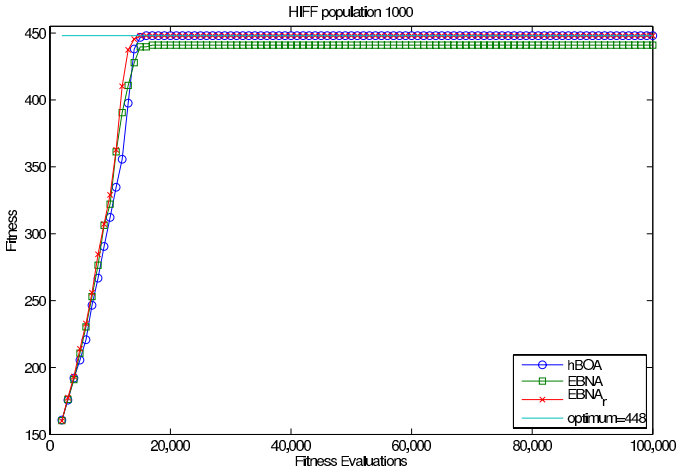

Fig. 7. The 64-bit HIFF problem. Comparison of best fitness using EBNA, hBOA and EBNA$_r$. Population size is 1000 individuals.

For the EqualProducts function, as we can see in Table II, in both cases the EBNA$_r$ model is the one which obtains a better result and besides with 1000 individuals the improvement is statistically significant.

TABLE II
RESULTS FOR THE FUNCTION EQUALPRODUCTS WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	evals. to best	total evals.	succeed
EBNA	○-2.80	47107	100000	0
hBOA	○-2.28	56553	100000	0
EBNA$_r$	●-2.24	60332	100000	0

(a)

EBNA	-1.07	66762	200000	0
hBOA	-1.23	104144	200000	0
EBNA$_r$	●-0.69	85768	200000	0

(b)

For the SixPeaks function the EBNA$_r$ model shows a weird behavior because with 500 individuals it has a higher number of succeeds than the other two models, but with 1000 individuals we can see that not only this model does not have more succeeds that the other two, is the worst, but besides it has less succeeds than with 500 individuals. This last characteristic also happends to EBNA so we can think that this fact is due to the probabilistic model and not to the diversity preservation scheme. Nonetheless EBNA$_r$ is better than EBNA in both cases.

TABLE III
RESULTS FOR THE FUNCTION SIXPEAKS WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	evals. to best	total evals.	succeed
EBNA	○75.82	17220	83120	11
hBOA	○80.58	26593	83250	14
EBNA$_r$	●83.46	20325	73410	17

(a)

EBNA	83.34	29305	169340	9
hBOA	●99.46	49324	84900	39
EBNA$_r$	88.40	35322	159020	13

(b)

For the knapsack problem, in Table IV there is almost no difference between the three models in any case, the three models should be considered as equally good with this function.

Last, for the HTRAP function we can see that EBNA$_r$ model is notably better than hBOA and EBNA. EBNA$_r$ reaches the optimum value in all cases, meanwhile hBOA and EBNA cannot do it in any case, and even EBNA$_r$ uses less evaluations to get the best value than hBOA.

Broadly speaking, after seeing these results for these five problems we can say that our proposed model first introduces an advance because in all cases it beats its base model, EBNA, and second because it seems to be competitive against hBOA, which is a reference algorithm in the field of EDAs for complex problems. However, as the proposed approach means to increase the diversity it is reasonable to think that our method can have worst performance in very simple problems. For that reason we tested the three models with the OneMax function, whose results are shown in Table VI. In this case, the problem is so simple that all models can reach the optimal solution in all runs, but EBNA needs less evaluation to do that. Nonetheless is very interesting to see that EBNA$_r$ only need a few more evaluations that EBNA while hBOA needs around double number of evaluations than EBNA.

TABLE IV
RESULTS FOR THE KNAPSACK PROBLEM WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	evals. to best	total evals.	succeed
EBNA	●602.00	26009	100000	0
hBOA	○601.98	14295	100000	0
EBNA$_r$	●602.00	25624	100000	0

(a)

EBNA	○601.96	42903	200000	0
hBOA	●602.00	28344	200000	0
EBNA$_r$	○601.96	29974	200000	0

(b)

TABLE V
RESULTS FOR THE FUNCTION HTRAP WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	eval. to best	total eval.	succeed
EBNA	47.47	5213	100000	0
hBOA	47.47	8102	100000	0
EBNA$_r$	●48.00	5983	6450	50

(a)

EBNA	47.47	9680	200000	0
hBOA	47.47	14121	200000	0
EBNA$_r$	●48.00	11050	11980	50

(b)

TABLE VI
RESULTS FOR THE FUNCTION ONEMAX WITH A POPULATION SIZE OF 500 (A) AND 1000 (B) INDIVIDUALS.

	fitness	eval. to best	total eval.	succeed
EBNA	100	5791	6130	50
hBOA	100	10915	11190	50
EBNA$_r$	100	6329	6630	50

(a)

EBNA	100	11541	12180	50
hBOA	100	33340	33880	50
EBNA$_r$	100	12786	13320	50

(b)

Given the use done in this work of the inverted probability distribution some questions and doubts could be arise over our proposal, specially with the results for the HTRAP function just due to the fact that in this function the local optimum and the global one are complementary strings. Nonetheless, taking into account the results for the other functions, which have not that peculiarity, this objection should be overcome.

VI. Conclusions

In this work we have studied one of the main problems that we can find in evolutionary algorithms, and in Estimation of Distribution Algorithms in particular, which is the premature convergence due to the lack of diversity necessary in the population. We have carried out an analysis about some proposals we can find in the literature and we have presented a new scheme which based on different premises.

The proposal presented here consists of trying to generate some individuals with the ability to break the homogeneity in the population, and to do that these individuals can not be generated just with the learned probability distribution. In stead we propose to use different pieces of information we have gathered about the model and the evolutionary process in order to generate a small percentage of individuals in the population with a different probability distribution than the learned one, although based on it, and something more elaborated than a simple mutation operator. The set of individuals generated in that way are called "rebels" and their goal is to lead the evolution to other areas of the search space in order to be able to overcome local optima points.

This proposal can be used with any probabilistic model since it only deals with the sampling process and then it can be applied to univariate, bivariate or multivariate models.

Taking into account the experimental evaluation we can conclude that this new scheme, in spite of being a first step in this line, shows very good results which back the use of this scheme and moreover justify the research of new variants more elaborated in order to improve results even more.

As future work we plan to study this proposal with discrete and continuous problems. Besides in these cases we have more choices to deal with the inverted probability distribution. Also we plan to use additional information join with the learn probability distribution in order to design new guides in the generation of rebels individuals.

Acknowledgment

This work has been partially supported by Spanish Ministerio de Educación y Ciencia (TIN2007-67418-C03-01); Junta de Comunidades de Castilla-La Mancha (PBI-08-048) and FEDER funds.

References

[1] D. Whitley, "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best," in *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, pp. 116–121.

[2] M. Lozano, F. Herrera, and J. R. Cano, "Replacement strategies to preserve useful diversity in steady-state genetic algorithms," *Information Sciences*, vol. 23, no. 178, pp. 4421–4433, 2008.

[3] M. Pelikan, *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms. Series: Studies in Fuzziness and Soft Computing, Vol. 170*. Springer, 2005.

[4] S. W. Mahfoud, "Niching methods for genetic algorithms," Illinois Genetic Algorithms Laboratory (IlliGAL) - University of Illinois at Urbana-Champaign, Tech. Rep. 95001, 1995.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[6] K. A. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, 1975.

[7] M. Pelikan and D. E. Goldberg, "Escaping hierarchical traps with competent genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Francisco, California, USA: Morgan Kaufmann, 7-11 2001, pp. 511–518. [Online]. Available: http://www.illigal.uiuc.edu/hboa/hboa.ps

[8] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 2005, pp. 24–31.

[9] T. Miquélez, E. Bengoetxea, and P. L. naga, "Evolutionary computation based on bayesian classifiers," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 3, pp. 335–349, 2004.

[10] J. M. Peña, J. A. Lozano, and P. Larrañaga, "Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of bayesian networks," *Evolutionary Computation*, vol. 13, no. 1, pp. 43–66, 2005.

[11] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Proceedings of the VI Workshop on Parallel Problem Solving from Nature*, 2000, pp. 385–394.

[12] D. Wallin and C. Ryan, "On the diversity of diversity," in *IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, pp. 95–102.

[13] Q. Zhang, J. Sun and E. Tsang, "An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 10, pp. 192–200, 2005.

[14] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Combinatonal Optimization by Learning and Simulation of Bayesian Networks," in *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, 2000, pp. 343–352.

[15] G. E. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[16] R. A. Watson, G. Hornby, and J. B. Pollack, "Modeling building-block interdependency," in *Proceedings of the 5th International Conference on Parallel Problem Solving From Nature*, 1998, pp. 480–490.

[17] S. Baluja and S. Davies, "Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)*, 1997, pp. 30–38.

[18] D. Pisinger, "Core problems in knapsack algorithms," *Operations Research*, vol. 47, pp. 570–575, 1994.

[19] J. L. Mateo and L. de la Ossa, "LiO: an easy and flexible library of metaheuristics," Departamento de Sistemas Informáticos, Escuela Politécnica Superior de Albacete, Universidad de Castilla-La Mancha, Tech. Rep. DIAB-06-04-1, 2006.