A Robust and Efficient Evolutionary Algorithm based on Probabilistic Model

Caichang Ding
School of Computer Science, Yangtze University, Jingzhou 434023, China
Email: hamigua_ping@hotmail.com

Wenxiu Peng
School of Computer Science, Yangtze University, Jingzhou 434023, China
Email: dianxin_1999@qq.com

***Abstract*—Evolutionary algorithms commonly search for the best solutions by maintaining a population of individuals that evolves from one generation to the next. The evolution consists of selecting a set of individuals from the population and applying, to some subsets of it, recombination operators that create new solutions. In this paper, Estimation of distribution algorithms arise as an alternative to genetic algorithms. Instead of exchanging information between individuals through genetic operators, Estimation of distribution algorithms use machine learning methods to extract relevant features of the search space through the selected individuals of the population. The replacement of crossover and mutation operators by probabilistic models can bring some benefits. The most important benefit could be that the structural component of the probabilistic model can provide explicit information about the interactions among the variables used to codify the problem solutions.**

***Index Terms*—interaction, machine learning, optimization, probabilistic model.**

I. Introduction

Looking for the best solutions to problems is not only a fundamental task for the development of mankind but also seems to be inherent to natural processes, and researchers have been able to see this. Proof of this is the emergence of evolutionary algorithms (EAs) to solve optimization problems regardless of the domain of application. This type of algorithms is mainly inspired by the way in which, according to Darwin [1], the adaptation of species to the environment is accomplished by nature. Nonetheless, other sources of inspiration from nature, such as the behaviors of ant colonies [2] or swarms [3], have also motivated the development of different EAs. Reciprocally, besides the inspiration of algorithms through the observation of nature, the study of such algorithms could provide us with a better understanding of nature.

EAs commonly search for the best solutions by maintaining a population of individuals (solutions) that evolves from one generation to the next. The evolution consists of selecting a set of individuals from the population and applying, to some subsets of it, recombination operators that create new solutions. A huge number of methods conforming to this framework have been developed. Therefore, the choice of the appropriate alternative for a particular application results in an important matter, as it may determine whether the problem is solved efficiently or, even, if the best solution is found at all.

Mathematically, optimization is the minimization or maximization of a given function. Hence, optimization problems can be formulated as,

$$x^* = \arg\max_x f(x), \tag{1}$$

where $f : S \to \mathrm{R}$ is called the objective function or fitness function, $x = (x_1, \ldots, x_n) \in S$ represents a possible solution of the problem and S is called the search space. The optimum x^* is not necessarily unique. We will assume that S is an n-dimensional discrete search space.

This paper is devoted to study a relatively new class of EAs: Estimation of distribution algorithms (EDAs) [4]. Based on the same principles of natural selection and evolution of populations, EDAs use explicit probability distributions to lead the search to promising areas of the search space instead of applying genetic operators of crossover and mutation used in genetic algorithms [5]. Throughout the paper, we will try to shed light on different open issues regarding EDAs. The final motivation is essentially to achieve a deeper understanding of this type of algorithms and their relationship with the optimization problems. To this end, novel methodological approaches and analyses have been conducted. The basic questions that have guided the elaboration of this work can be summarized as follows.

Firstly, the learning of probabilistic models to extract the relevant information that the selected individuals can contain about the problem is a fundamental step of the algorithm. Regarding this issue, we wonder how the search and the behavior of the EDA is influenced by the accuracy of the learning method.

Secondly, one of the most interesting properties of EDAs is their ability to capture and explicitly represent

doi:10.4304/jcp.9.6.1462-1469

interactions among the variables of the problems by means of the probabilistic models. Thus, investigating the relationship between the interactions of the problem variables and the structure of the probabilistic model is a question that arises naturally. Following this idea, we also wonder how the topology that these interactions provide determines the difficulty of the problem. More generally, the question of what makes a problem difficult for EDAs is an open question of undoubted interest.

Thirdly, a utopian goal is to know the limits of effectiveness of any search algorithm. Among other things, this type of knowledge would allow us to select the most adequate algorithm depending on the problem at hand. Coming back to more affordable issues, we wonder where the learning limits of EDAs are. We want to better understand when and why the learning step is not able to extract from the population the needed information to reach the optimum.

Fourthly, another fundamental issue regarding EDAs that we consider of special interest is to better understand how the probability of the optimum evolves during the generations. This is an essential characteristic of the algorithm which reflects how the problem is being solved. And finally, a more general issue that we keep in mind is the relationship that emerges between an EDA and the space of optimization problems. Regarding this issue, we consider the possibility of creating taxonomies of problems according to the different behaviors that an EDA can exhibit.

This introductory part will treat, as directly and briefly as possible, the theoretical background related with the paper. Thus, only Bayesian networks and EDAs are formally presented. Further details of any topic or scientific discipline related with the aforementioned elements, but not directly used throughout the paper, can be consulted in different works that will be cited in the appropriate places. In turn, the specific theoretical background that the different chapters could need, will be introduced in the corresponding points.

The rest of this paper is organized as follows. Estimation of distribution algorithms are introduced in section II. Section III presents Bayesian networks. Section IV discusses the parameters of the EDAs. At last, the summary is given in section V.

II. Estimation of Distribution Algorithms

Estimation of distribution algorithms [4][6] are a population-based optimization paradigm in the field of evolutionary computation [7]. Initially, a random sample of solutions is generated. These solutions are evaluated using the objective function, and a subset of candidate solutions is selected based on this evaluation. Hence, solutions with better function values have a higher chance of being selected. Then, a probabilistic model from the selected set is built and a new population is sampled from that model. The process is iterated until the optimum has been found or another termination criterion is fulfilled. The general scheme of the EDA approach is shown in Figure 1.

EDAs arise, in part, as an alternative to genetic algorithms [5]. Instead of exchanging information between individuals through genetic operators, EDAs use machine learning methods to extract relevant features of the search space through the selected individuals of the population. The replacement of crossover and mutation operators by probabilistic models can bring some benefits. For example, EDAs reduce the number of parameters involved and hence, the tune of the algorithm could become simpler depending on the scenario of application. Nevertheless, the most important benefit could be that the structural component of the probabilistic model can provide explicit information about the interactions among the variables used to codify the problem solutions.

$D_{t=0} \leftarrow$ Generate N individuals randomly
do
  $D_t \leftarrow$ Evaluate individuals
  $D_t^{Se} \leftarrow$ Select $M \leq N$ individuals from D_t according to a selection method
  $p_t(\boldsymbol{x}) = p(\boldsymbol{x}|D_t^{Se}) \leftarrow$ Estimate the joint probability distribution by means of a probabilistic model
  $D_{t+1} \leftarrow$ Sample M individuals from $p_t(\boldsymbol{x})$ and create the new population
  $t \leftarrow t+1$
until Stopping criterion is met

Figure 1. The general scheme of estimation of distribution algorithms.

With the aim of finding the optimal solution x^* and solving Problem in (1), EDAs use explicit probability distributions. At each iteration, the algorithm manages a probability distribution $p(X = x)$ of the random variable X taking values from the search space S. Thus, each of the possible problem solutions has an associated probability of being sampled which varies during the optimization process. The probability values assigned to the solutions are the main source in determining which one will be returned by the algorithm. Consequently, given a problem, the main goal is to get higher probability values for the highest quality solutions throughout an iterative process.

In the last decade, EDAs have acquired special relevance. Proof of this popularity is the development of new and more complex EDAs [8][9], the applications for these EDAs in different domains such as engineering [10], biomedical informatics or robotics [11] and the works which study fundamental issues in order to better understand how these algorithms perform [12].

Although there is a wide variety of EDA implementations, as an example, we present below the pseudocode of the univariate marginal distribution algorithm (UMDA), the tree-based estimation of distribution algorithm (Tree- EDA) and the estimation of Bayesian networks algorithm (EBNA). These algorithms will be considered in subsequent sections of the paper.

A. Univariate Marginal Distribution Algorithm

The univariate marginal distribution algorithm was introduced in [13]. This algorithm assumes that all the variables are independent. That is, the value of variable

X_i does not depend on the state of any other variable. Then, $p(x)$ can be factorized as follows:

$$p(x) = \prod_{i=1}^{n} p(x_i) \quad (2)$$

Figure 2 shows the steps of the UMDA. This algorithm has been successfully applied to different problems such as feature subset selection [14], learning of Bayesian networks from data [15], optimization of a composite video processing system [16], or to solve some linear and combinatorial problems using Laplace correction [17].

Theoretical results derived from the UMDA [4] expose its relationship with GAs, particularly with GAs that use uniform crossover. [18] have investigated some of the issues that explain the success of UMDA in the optimization of a wide class of functions. Other theoretical results have been obtained for UMDA in [19].

$D_{t=0} \leftarrow$ Generate N individuals randomly
do
  $D_t \leftarrow$ Evaluate individuals
  $D_t^{Se} \leftarrow$ Select $M \leq N$ individuals from D_t according to a selection method
  Calculate the univariate marginal frequencies $p_t^s(x_i)$
  $D_{t+1} \leftarrow$ Sample N individuals from $p_t(\boldsymbol{x}) = \prod_{i=1}^{n} p_t^s(x_i)$
  $t \leftarrow t+1$
until Stopping criterion is met

Figure 2. Pseudocode for UMDA.

B. Tree-based Estimation of Distribution Algorithms

Tree-based estimation of distribution algorithms [20] use factorizations that can be expressed by means of trees or forests. In particular, we will focus on the implementation. The pseudocode of this algorithm is shown in Figure 3 and will be called Tree-EDA. Although other methods can also be employed, the factorization is constructed using the algorithm introduced in [21] that calculates the maximum weight spanning tree from the matrix of mutual information between pairs of variables. Additionally, a threshold for the mutual information values is used when calculating the maximum weight spanning tree in order to allow disconnected components in the structural model.

$D_{t=0} \leftarrow$ Generate N individuals randomly
do
  $D_t \leftarrow$ Evaluate individuals
  $D_t^{Se} \leftarrow$ Select $M \leq N$ individuals from D_t according to a selection method
  Calculate the univariate and bivariate marginal frequencies $p_t^s(x_i)$ and $p_t^s(x_i, x_j)$ from D_t^{Se}
  Calculate the mutual information and learn the tree structure
  $D_{t+1} \leftarrow$ Sample N individuals from the tree
  $t \leftarrow t+1$
until Stopping criterion is met

Figure 3. Pseudocode for Tree-EDA.

C. EDAs based on Bayesian Networks

Throughout the paper, we pay special attention to EDAs that learn Bayesian networks. There are different implementations of this type of EDAs. The best known algorithms could be the following, such as learning factorized distribution algorithm (LFDA), Bayesian optimization algorithm (BOA) or estimation of Bayesian networks algorithm (EBNA). We mainly focus on the EBNA implementation whose pseudocode is presented in Figure 4.

$BN_{t=0} \leftarrow (\boldsymbol{s}_0, \boldsymbol{\theta}_{\boldsymbol{s}_0}^0)$ where $\boldsymbol{s}_0$ is an arc-less structure and $\boldsymbol{\theta}_{\boldsymbol{s}_0}^0$ is uniform
$D_{t=0} \leftarrow$ Generate N individuals from BN_0
do
  $D_t \leftarrow$ Evaluate individuals
  $D_t^{Se} \leftarrow$ Select $M \leq N$ individuals from D_t according to a selection method
  $\boldsymbol{s}_t \leftarrow$ Obtain a network structure
  $\boldsymbol{\theta}^t \leftarrow$ Calculate θ_{ijk}^t using D_t^{Se} as the data set
  $BN_t \leftarrow (\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{s}_t}^t)$
  $D_{t+1} \leftarrow$ Sample N individuals from BN_t and create the new population
  $t \leftarrow t+1$
until Stop criterion is met

Figure 4. Pseudocode for EBNA

In order to better understand how EDAs based on Bayesian networks perform, the characteristics of the learned probabilistic models are a rich source of information which has been studied in several works [22-25]. A straightforward form of analysis is through the explicit dependences between the variables they capture. Thus, it has been shown how different parameters of the algorithm influence the accuracy of the structural models [24], how the dependencies of the probabilistic models change during the search and, how the networks learned can provide information about the problem structure [23]. Moreover, the structural component of the model can be used to introduce available information of the structural characteristics of the problem [26].

III. Bayesian Networks

All the algorithms considered throughout the paper use factorizations that can be encoded by means of Bayesian networks . Bayesian networks, also called belief networks, are a class of probabilistic graphical model. This type of models have become a very popular paradigm to efficiently deal with probability distributions in modeling uncertain knowledge. One of the most important sources of the development of Bayesian networks was the field of expert systems. In addition, over the last few years, Bayesian networks have received considerable attention from the machine learning community. As a result of this interest, many publications and tutorials have appeared. Thus, besides expert systems, the applications of Bayesian networks include classification problems, optimization or bioinformatics.

As any other probabilistic graphical model, Bayesian networks are the result of combining probability and graph theory. The graphical component of the model encodes a list of conditional independences [27-28]

associated to the probability distribution. Let $X=(X_1,\ldots,X_n)$ be an n-dimensional discrete random variable. A Bayesian network is a graphical representation of the factorization of the joint probability distribution for X, $p(X=x)$, where $x=(x_1,\ldots x_n)$ denotes an assignment of the variable X. More specifically, a Bayesian network can be defined as a pair (s,θ_s) where s is a directed acyclic graph (model structure) and θ_s is the set of parameters associated to the structure (model parameters). The structure s determines the set of conditional (in)dependences among the random variables of X. According to the structure s, the joint probability distribution $p(x)$ can be factorized by means of marginal and conditional probability functions. Specifically, the probability distribution factorizes according to the graph as,

$$p(x)=\prod_{i=1}^{n} p(x_i \mid pa_i) \tag{3}$$

where pa_i denotes a value of the variables Pa_i, the parent set of X_i in the graph s.

The local probability distributions of the factorization are those induced by the terms of the product that appears in (3). We assume that these local distributions depend on the parameters $\theta_s=(\theta_1,\ldots,\theta_n)$. Equation (3) can be rewritten specifying the parameters:

$$p(x \mid \theta_s)=\prod_{i=1}^{n} p(x_i \mid pa_i,\theta_i) \tag{4}$$

Assuming that the variable X_i has r_i possible values, the local distribution $p(x_i \mid pa_i^j,\theta_i)$ is an unrestricted discrete distribution:

$$p(x_i^k \mid pa_i^j,\theta_i)=\theta_{ijk} \tag{5}$$

where $pa_i^1,\ldots,pa_i^{q_i}$ denote the q_i possible values of the parent set Pa_i. The parameter θ_{ijk} represents the probability of variable X_i being in its k-th value, knowing that the set of its parents' variables is in its j-th value. Therefore, the local parameters are given by $\theta_i=((\theta_{ijk})_{k=1}^{r_i})_{j=1}^{q_i}$.

A. Bayesian Network Learning

In order to obtain a Bayesian network which allows us to represent and manage the uncertain knowledge of a specific domain, it is necessary to set both the structure and the parameters. The structure and conditional probabilities necessary for characterizing the Bayesian network can be provided either externally by experts, by automatic learning from datasets or by mixing both of these. We focus on the second approach. Moreover, when the model is automatically learned, it can provide us with insights into the interactions between the variables of the domain.

The learning task can be separated into two subtasks: structural learning and parameter learning. Although there are different strategies to learn the structure of a Bayesian network, we focus on the so-called score+search approach. This type of techniques deals with the structure learning as an optimization problem. Therefore, learning a Bayesian network can be enunciated as follows. Given a data set D with N cases, $D=\{x_1,\ldots,x_N\}$, searching the structure s^* such that,

$$s^*=\arg\max_{s\in S^n} g(s,D) \tag{6}$$

where $g(s,D)$ is the score or metric which measures the goodness of any given structure s with respect to the data set D, and S_n is the set of all possible directed acyclic graphs with n nodes. Some of the most relevant and used heuristic techniques such as greedy search, simulated annealing, genetic algorithms, estimation of distribution algorithms or ant colony optimization have been applied to this task.

One of the desirable properties of a metric or score is the decomposability in presence of complete data sets. These metrics can be decomposed in sub-metrics associated to each node X_i and its parents Pa_i in the graph s. Formally, any decomposable metric can be expressed as:

$$g(s,D)=\sum_{i=1}^{n} g_D(X_i,Pa_i) \tag{7}$$

where the function g_D is the sub-metric. Due to the decomposability, the local search methods are computationally more efficient because after adding an arc, we only need to evaluate the family of nodes affected by this change.

Although different learning methods are considered throughout the paper, a specific search algorithm will be generally used. It is Algorithm B [29]. This is a greedy search algorithm and the pseudocode is presented in Figure 5, where A is a data structure that stores the information needed to manage the addition of the candidate arcs. Basically, Algorithm B starts with an arcless structure and, at each step, adds the arc which improves the score the most. The algorithm finishes when there is no arc whose addition improves the score.

Start with an arcless structure
Compute $A[X_j \to X_i] = g_D(X_i, X_j) - g_D(X_i)$ for all distinct X_i, X_j
do
  Look for the largest $A[X_j \to X_i]$ and add that arc $X_j \to X_i$ to $\boldsymbol{s}$
  $A[X_j \to X_i] = g_D(X_i, \boldsymbol{Pa}_i \bigcup X_j) - g_D(X_i, \boldsymbol{Pa}_i)$ for all distinct X_i, X_j not belonging to $\boldsymbol{Pa}_i$
  $A[X_j \to X_i] = -\infty$
until Every $A[X_j \to X_i] < 0$

Figure 5. Pseudocode for Algorith B

Regarding the implementation of the score $g(s, D)$, different alternatives can be considered. Among the most used families of scores we can find marginal likelihood, penalized log-likelihood or information theory based scores. In the current section we will use the Bayesian Information Criterion score (BIC) [30] based on penalized maximum likelihood. This metric is obtained as follows. Given a dataset $D = \{x_1, \ldots, x_N\}$, we might calculate for any Bayesian network structure s the maximum likelihood estimate $\hat{\theta}_s$ for the parameters θ_s and the associated maximized log likelihood:

$$\begin{aligned} \log p(D \mid s, \theta_s) &= \log \prod_{w=1}^{N} p(x_w \mid s, \theta_s) \\ &= \log \prod_{w=1}^{N} \prod_{i=1}^{n} p(x_{w,i} \mid pa_i, \theta_i) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \log(\theta_{ijk})^{N_{ijk}} \end{aligned} \quad (8)$$

where N_{ijk} denotes the number of cases in D in which the variable X_i has the value x_i^k and Pa_i has its j-th value. Since the maximum likelihood estimate for θ_{ijk} is given by $\hat{\theta} = \frac{N_{ijk}}{N_{ij}}$ where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, we obtain:

$$\log p(D \mid s, \hat{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} \quad (9)$$

The log-likelihood function is not used to guide the search process due to two main problems. Firstly, the log-likelihood is a monotonous increasing function with respect to the complexity of the model structure. Therefore, the use of this score to evaluate the quality of the structures during the search could lead us towards complete Bayesian networks. Secondly, as the number of parameters for each node increases, the error in the parameter estimation also increases. In order to overcome these difficulties, a penalty term is added to the log-likelihood. A general formula of the penalized log-likelihood is given by:

$$\sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - h(N) dim(S) \quad (10)$$

where $dim(S)$ is the dimension (number of parameters needed to specify the model) of the Bayesian network with a structure s, i.e. $dim(S) = q_i(r_i - 1)$. $h(N)$ is a non-negative penalization function. The Jeffreys- Schwarz criterion, which is usually called BIC [30], takes into account $h(N) = \frac{1}{2} \log N$. Thus, the BIC score can be written as follows:

$$\begin{aligned} &BIC(s, D) \\ &= \log \prod_{w=1}^{N} \prod_{i=1}^{n} p(x_{w,i} \mid pa_i, \hat{\theta}_i) - \frac{1}{2} \log N \sum_{i=1}^{n} q_i (r_i - 1) \end{aligned} \quad (11)$$

On the other hand, parameter learning is the numerical assessment of the parameters θ_S that specify the conditional and marginal probability distributions of the factorization given by s. Although this task can be done by means of different approaches such as the Bayesian model averaging or the maximum a posteriori criterion [31], we use the maximum likelihood estimation. Specifically, once the structure has been learned, the parameters of the Bayesian network are calculated by using the Laplace correction as follows:

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + 1}{N_{ij} + r_i} \quad (12)$$

B. Simulation

Once a Bayesian network is obtained, this model is able to provide us with specific probabilistic information of interest. Usually, the information that the practitioner wants to know is the probability of a certain event in the light of particular observations or evidence. The probabilities of interest are not usually stored in the Bayesian network at hand, they need to be computed. This process is known as probabilistic inference and, in the general case, it is an NP-complete problem [32].

Simulation (also called stochastic sampling) of Bayesian networks can be considered as an alternative to the exact inference. The simulation of any probabilistic graphical model consists of obtaining a sample from the probability distribution for X that the model encodes. Then, the marginal or conditional probabilities of interest can be estimated from the sample.

For our purposes regarding EDAs, the objective of the simulation of Bayesian networks is to obtain a dataset (new population) in which the probabilistic relationships between the random variables of the model are underlying. Particularly, in order to sample the Bayesian network, we consider a forward sampling method. A variable is sampled once all its parents have been sampled. This method is known as probabilistic logic

sampling (PLS). Figure 6 shows a pseudocode of this method.

$\pi \leftarrow$ Ancestral ordering of the nodes in the Bayesian network
for $j = 1$ to N
  for $i = 1$ to n
    $x_{j,\pi(i)} \leftarrow$ Randomly generate a value from $p(x_{\pi(i)}|pa_{\pi(i)})$
  done
done

Figure 6. Pseudocode of the probabilistic logic sampling method.

IV. Parameters of the EDAs

We have set a configuration of the EDA parameters which is often used throughout the paper. Therefore, this standard configuration is introduced here to avoid unnecessary repetitions.

According to the main scheme of the EDA, it works with populations of N individuals. The initial population is generated according to a uniform distribution, and hence, all the solutions have the same probability of being sampled. Each iteration starts by selecting a subset of promising individuals from the population. In this step we use truncation selection with a threshold of 50%. Thus, the $N/2$ individuals with the best fitness value are selected. The next step is to learn a probabilistic model from the subset of selected individuals. This is the only step where the algorithms that we will consider differ. Once the model is built, the new population can be generated. In order to do that, N new solutions are sampled from the probabilistic model and then they are added to the N individuals of the current population. The N best individuals, among the $2N$ available, constitute the new population.

As previously commented, every EDA considered in the paper uses factorizations that can be encoded by means of Bayesian networks. Therefore, the same approaches can be used both to obtain the corresponding parameters and to sample the new solutions. As explained above, the parameters are estimated by maximum likelihood and the new population is generated by PLS (see Figure 6).

V. Summary

This paper has been devoted to increase our comprehension about EDAs.

The relationship between the structure that the interactions of the problem variables provide and the structural models learned by the algorithm has been a issue. In this regard, we have seen that the structures that the algorithm learns during the search provide valuable information about the interdependences among the variables of the problem. This fact has been observed in other related works and it is considered as a distinctive feature of EDAs compared with other types of evolutionary algorithms. However, it has also been noticed that introducing a learning method that obtains the best Bayesian networks at each generation does not necessarily improve the performance of the algorithm. Nevertheless, with enough population size, this type of algorithm is able to obtain structures that provide much more information about the problem than the approximate learning.

When the algorithm is studied from the perspective of the probability of the optimum and the most probable solution, novel insights can be provided. The main elements of the algorithm that we have considered, which are the structural model and the population size, clearly influence the probability of the optimum and the most probable solution. Moreover, the patterns of behavior are constant in every optimization problem analyzed. For instance, using an adequate population size or an accurate structural model increases the probability of the optimum during the search in relation to the most probable solution, even in runs where the optimum is not reached. In addition, the function values of the most probable solution also reflect the influence of the population size and the structural model accuracy. The properties of the problem at hand, such as the multimodality, or even the difficulty that it entails for the algorithm, are reflected in this type of analysis. The experimental framework designed is not only useful to better understand EDAs but also to devise new improvements of the algorithm.

As previously commented, the relationship between the structure of the problem and the structural models used by EDAs is a issue. In this regard, different adjectives such as benign, malign, strong or deceptive have been used to describe the interactions among the variables of the problem and then, study their effect both in EDAs and other evolutionary algorithms. Although some attempts to formalize this type of concepts have been presented, we clearly need to conduct more research in order to understand and specify all the aforementioned terms in the context of optimization by means of EDAs.

Regarding the limits of effectiveness in EDAs, a more in-depth study should be carried out in order to increase the soundness of the conclusions. Thus, more accurate learning techniques, more sophisticated EDAs aided by niching or local searches, or even other approaches such as mixtures of evolutionary algorithms, should be tested under the same worst-case scenario. Then, analyzing the levels of problem difficulty that this type of algorithms successfully reaches, would be useful to better understand both the learning limits of EDAs and the limits of other search techniques. To complement the results obtained by using functions based on deceptive sub-functions, similar experiments could be conducted with other classes of functions such as Max-SAT or Ising. The role of the population in the limits of effectiveness of the algorithm was also discussed. We argue that a given population size can only contain useful information to solve problems to a certain degree of interaction among their variables. However, studies related with the information that the populations contain about the problem have hardly been considered. We believe that the formalization and study of this notion would be worthwhile.

The taxonomy of problems opens new research lines. First of all, some generalization such as the introduction of non-injective functions and general Bayesian networks could be developed. In addition, providing the needed definitions to deal with any type of selection scheme could also be considered. Other important extensions are related to the connection between the characteristics of the problems and the equivalence classes to which they belong. We have shown the connection of the classes with the neighborhood system induced by the Hamming distance for univariate EDAs. This connection can be studied for more complex probabilistic models. For example, preliminary results indicate that, if we add an arc to the univariate model, then it is possible to include functions with one and two local optima in the same class. This implies that some functions with two local optima can entail the same difficulty as functions with one local optimum (the global optimum). This agrees with that using higher order statistics could improve the chance of finding the global optimum. Moreover, we hypothesize that it is possible to discover new links with other problem characteristics or descriptors. For instance, we have very preliminary results regarding the additive decomposition of the functions and its relationship with the equivalence classes. In turn, the classes could also be tagged in terms of the difficulty of the problems they contain. In an ideal scenario, the information available about the problem at hand could be used to identify the class to which it belongs to and then try to advance, for example, whether for a given factorization the algorithm will reach the optimum. In fact, knowing if a determined factorization will converge to the optimum for a given function is one of the most important issues in EDAs.

ACKNOWLEDGMENT

This paper is supported in part by National Natural Science Foundation of China(Grant no.60975050), Research Fund for the Doctoral Program of Higher Education(Grant no.20070486081) and Fundamental Research Funds for the Central Universities(Grant no.6081014). The authors are grateful to the four anonymous referees for their insightful and constructive comments, which greatly improved the quality of the paper.

REFERENCES

[1] C. Darwin, The Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life, 1859.

[2] M. Dorigo, and T. Stützle, Ant Colony Optimization. MIT press, 2004.

[3] J. Kennedy, and R. Eberhart, Particle swarmoptimization. In Proceedings of the IEEE International Conference on Neural Networks, volume 4, pp. 1942–1948. IEEE Press, 1995.

[4] H. Mühlenbein, and G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters. In Voigt, H.-M., Ebeling, W., Rechenberg, I., and Schwefel, H.-P., editors, Parallel Problem Solving from Nature (PPSN IV), volume 1141 of Lectures Notes in Computer Science, pp. 178-187, Berlin. Springer Verlag, 1996.

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

[6] P. Larrañaga, and J. A. Lozano, editors, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[7] E. A. Eiben, and J. E. Smith, Introduction to Evolutionary Computing (Natural Computing Series). Springer, 2003.

[8] P. A. Bosman, The Anticipated Mean Shift And Cluster Registration In Mixture-Based EDAs For Multi-Objective Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010), pp. 351 – 358. ACM Press, 2010.

[9] M. Hauschild, M. Pelikan, K. Sastry, , and Goldberg, D. E. (2012). Using previous models to bias structural learning in the hierarchical BOA. Evolutionary Computation, 20(1): pp. 135–160, 2012.

[10] P. A. Simionescu, D. Beale and G. V. Dozier, Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm. Journal of Mechanical Design, 128(1): pp. 108–115, 2007.

[11] B. Yuan, M. E. Orlowska and S. W. Sadiq, Finding the optimal path in 3D spaces using EDAs - the wireless sensor networks scenario. In Proceedings of the Adaptive and Natural Computing Algorithms, 8th International Conference (ICANNGA-2007), pages 536–545, Warsaw, Poland. Springer Verlag, 2007.

[12] J. L. Shapiro, Drift and scaling in estimation of distribution algorithms. Evolutionary Computation, 13(1): pp. 99–123, 2005.

[13] H. Mühlenbein, The equation for response to selection and its use for prediction. Evolutionary Computation, 5(3): pp. 303–346, 1998.

[14] R. Blanco, P. Larrañaga, I. Inza and B. Sierra, Selection of highly accurate genes for cancer classification by estimation of distribution algorithms. In Proceedings of the Workshop Bayesian Models in Medicine held within (AIME-2001), pp. 29-34, 2001.

[15] R. Blanco, I. Inza and P. Larrañaga, Learning Bayesian networks in the space of structures by estimation of distribution algorithms. International Journal of Intelligent Systems, 18(2): pp. 205-220, 2003.

[16] W. Ali and A. P. Topchy, Memetic optimization of video chain designs. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004), pp. 869–882, Seattle, WA, USA. Springer, 2004.

[17] T. Paul and H. Iba, Linear and combinatorial optimizations by estimation of distribution algorithms. In Proceedings of the 9th MPS Symposium on Evolutionary Computation, pp. 99–106, 2003.

[18] H. Mühlenbein and T. Mahnig, Evolutionary computation and beyond. In Y. Uesaka, P. Kanerva and H. Asoh, Eds, Foundations of Real-World Intelligence, pp. 123–188. CSLI Publications, Stanford, California, 2001.

[19] C. González, J. A. Lozano and P. Larrañaga, Mathematical modeling of UMDAc algorithm with tournament selection. Behaviour on linear and quadratic functions. International Journal of Approximate Reasoning, 31(4): pp. 313–340, 2002.

[20] S. Baluja and S. Davies, Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In Proceedings of the 14th International Conference on Machine Learning, pp. 30–38. Morgan Kaufmann, 1997.

[21] C. K. Chow and C. N. Liu, Approximating discrete probability distributions with dependence trees. IEEE

Transactions on Information Theory, 14(3): pp. 462–467, 1968.
[22] M. Hauschild, and M. Pelikan, Enhancing efficiency of hierarchical BOA via distance-based model restrictions. MEDAL Report No. 2008007, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), 2008.
[23] M. Hauschild, M. Pelikan, K. Sastry, and C. Lima, Analyzing Probabilistic Models in Hierarchical BOA. IEEE Transactions on Evolutionary Computation, 13(6): pp. 1199–1217, 2009.
[24] C. F. Lima, M. Pelikan, D. E. Goldberg, F. G. Lobo, K. Sastry and M. Hauschild, Influence of selection and replacement strategies on linkage learning in BOA. In Proceedings of the 2007 Congress on Evolutionary Computation (CEC-2007), pp. 1083–1090. IEEE Press, 2007.
[25] H. Mühlenbein and R. Hőns, The factorized distributions and the minimum relative entropy principle. In Pelikan, M., Sastry, K., and Cantú-Paz, E., editors, Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications, Studies in Computational Intelligence, pp. 11–38. Springer-Verlag, 2006.
[26] M. Hauschild, M. Pelikan, K. Sastry and D. E. Goldberg, Using previous models to bias structural learning in the hierarchical BOA. Evolutionary Computation, 20(1): pp. 135–160, 2012.
[27] A. P. Dawid, Conditional independence in statistical theory. Journal of the Royal Statistical Society Series B, 41: pp. 1–31, 1979.
[28] A. P. Dawid, Conditional independence for statistical operations. Annals of Statistics, 8(3): pp. 598–617, 1980.
[29] W. Buntine, Theory refinement on Bayesian networks. In Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, pp. 52–60, 1991.
[30] G. Schwarz, Estimating the dimension of a model. Annals of Statistics, 7(2): pp. 461–464, 1978.
[31] D. Heckerman, D. Geiger and D. M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 20: pp. 197–243, 1995.
[32] G. F. Cooper, The computational complexity of probabilistic inference using belief networks. Artificial Intelligence, 42: pp. 393–405, 1990.

Caichang Ding received the B.Sc. degree from the School of Mechanical & Electronic Information, China University of Geosciences, Wuhan, China, in 2003, and the M.Sc. degree from the School of Computer, Wuhan University, Wuhan, China, in 2006. He is currently a lecturer in the School of Computer Science, Yangtze University, Jingzhou, China. His main research interests include computational learning theory, statistical learning, basic theory of evolutionary computation and optimization theory.

Wenxiu Peng received the B.Sc. and M.Sc. degrees from Hubei University, Hubei, China, in 2003 and 2006. She is currently a lecturer in the School of Computer Science, Yangtze University, Jingzhou, China. Her main research interests include computational learning theory, statistical learning, basic theory of evolutionary computation and optimization theory.