



Univariate marginal distribution algorithm dynamics for a class of parametric functions with unitation constraints

Li-Vang Lozada-Chang^a, Roberto Santana^{b,c,*}

^a Faculty of Mathematics and Computation, University of Havana, San Lázaro y L, CP-10400 La Habana, Cuba

^b Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizábal 1, CP-20080 San Sebastián -Donostia, Spain

^c Universidad Politécnica de Madrid, Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid, Spain

ARTICLE INFO

Article history:

Received 26 March 2007

Received in revised form 15 November 2010

Accepted 13 January 2011

Available online 21 January 2011

Keywords:

Estimation of distribution algorithms

Univariate marginal distribution algorithm

Convergence analysis

Evolutionary algorithms

Long string limit analysis

ABSTRACT

In this paper, we introduce a mathematical model for analyzing the dynamics of the univariate marginal distribution algorithm (UMDA) for a class of parametric functions with isolated global optima. We prove a number of results that are used to model the evolution of UMDA probability distributions for this class of functions. We show that a theoretical analysis can assess the effect of the function parameters on the convergence and rate of convergence of UMDA. We also introduce for the first time a long string limit analysis of UMDA. Finally, we relate the results to ongoing research on the application of the estimation of distribution algorithms for problems with unitation constraints.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Evolutionary algorithms that use probabilistic models instead of genetic operators have been praised for their ability to solve problems quickly, reliably and accurately [25]. These are usually called estimation of distribution algorithms (EDAs) [12,22]. EDAs are non-deterministic heuristic search strategies that use a population of solutions instead of a single point. In every iteration, usually called a generation, a subset of solutions is selected, and a probabilistic model of these solutions is constructed. A new population of points is generated from this model. The algorithm iterates (evolves) until a stopping condition is satisfied.

The probabilistic model that EDAs construct in each generation is designed to capture a number of relevant relationships in the form of statistical dependencies between the variables. These dependencies are then used to generate solutions that must share a number of characteristics with the selected solutions. In this way, the search is led to the most promising areas of the search space. It has been shown that this strategy can be highly effective for dealing with problems where traditional genetic algorithms (GAs) fail (e.g., deceptive functions) [12,21,25].

Another key feature of EDAs is their suitability for theoretical analysis. This type of study has been widely pursued using facetwise models [6]. Facetwise models consider models for various effects separately. These models have been used to study population sizing and mixing and to conduct convergence-time analyse [23].

* Corresponding author at: Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizábal 1, CP-20080 San Sebastián -Donostia, Spain.

E-mail addresses: livang@gmail.com (L.-V. Lozada-Chang), roberto.santana@upm.es (R. Santana).

Based on the theories of graphical models and probability estimation, a number of mathematical approaches to EDA dynamics are possible. Early mathematical analyses of EDAs showed the relationship between these algorithms and traditional GAs [3,15,22]. Current theoretical analyses of EDAs include proofs of convergence for different EDAs [8,21,38,41]. Results have also been reported on minimum population size bounds and the number of generations to convergence [23,24] as well as dynamics analysis of some EDAs for certain classes of functions [7,14,18,40]. Recently, work on the mathematical modeling of EDAs has been divided into two classes: (1) analysis based on the theory of Markov models, and (2) analysis based on dynamical systems theory. The work presented in this paper falls into this second class.

We will focus on UMDA [17], an EDA that uses a factorized probability model based on the univariate marginals calculated from the selected population. Although UMDA has already been analyzed to some extent, its study is still a key topic in the field of EDAs on theoretical and practical grounds.

From a theoretical point of view, it was recognized early on [16,22] that the close relationship between the dynamics exhibited by GAs and the UMDA justified the theoretically more amenable analysis of UMDA as a way to study the behavior of GAs.

Moreover, UMDA is a member of the class of EDAs that employ univariate probability models, such as the compact genetic algorithm [11] and the population-based incremental learning algorithm [2]. Theoretical analysis of UMDA can be applied to these algorithms. This could explain why, while EDAs using multivariate probabilistic models have begun to attract more attention in recent years [9,13], interest in analyzing the dynamics of UMDA for different classes of functions has not waned [35,40]. From a practical point of view, UMDA and other EDAs based on univariate models are a suitable alternative for many applications [4,10,34,36,37], particularly if memory and time constraints [30,42] are critical for the success of the optimization algorithms.

Mühlenbein et al. [19] have explained why UMDA is good for optimizing a wide class of functions. They state that the algorithm transforms the original fitness landscape defined by the objective function into the landscape of the average fitness determined by the probabilistic model. This transformation smoothes the rugged fitness landscape so that UMDA will find the global optimum if there is a tendency toward it.

We introduce a class of parametric functions whose members are functions of different levels of difficulty that are completely described by a set of parameters. The aim of this paper is to investigate UMDA performance for different members of the function class and how this performance is related to the parameters that determine each function landscape. To do this, we prove a number of results that describe the variation in UMDA dynamics for the different functions using probability distributions, and we show how the dynamics determine whether the algorithm can find the optimum.

In Section 2, we give a brief introduction to UMDA with proportional selection. In Section 3, we present the class of unification functions, and we obtain the equations that describe the UMDA dynamics for these functions. Later, we define a class of parametric unification functions, and we study the UMDA dynamics for the different parameters of this class of functions. In Section 4, we discuss the limits of the infinite population approximation and the implications of using finite populations instead. In Section 5, we explain how the results can be extended to more complex multimodal functions. The analysis of the UMDA dynamics for infinite populations reveals that the algorithm has a long string limit behavior. This result is formalized in Section 6. In Section 7, we briefly discuss a number of problems to which the results could be applied. The paper offers conclusions and a discussion of further work in Section 8.

2. UMDA

The search space Ω will be the ordinary n -dimensional binary space, i.e., $\Omega = \{0,1\}^n$. Let f be a non-negative real-valued function defined on Ω . The goal of evolutionary optimization algorithms such as UMDA is to find the global extrema of f . In this paper, we will assume that the problem is to find the maximum of f .

For instant t , let \mathbb{P}_t (resp. \mathbb{P}_t^S) be the probability of an individual being a member of the population (resp. the probability determined by a given selection method.) Let X_t (resp. X_t^S) denote the random vector of Ω with distribution \mathbb{P}_t (resp. \mathbb{P}_t^S). We will sometimes drop the subscript t and denote the above random vector by X (resp. X^S) when it is clear which is the instant t .

An ideal evolutionary algorithm should strike a fine balance between search space exploitation and exploration. The search can be thought of as a process comprising two stages. In the first stage, the algorithm learns an as-accurate-as-possible approximation of the selected population. In the second stage, this approximation is modified to allow for exploration. In practice, it is difficult to find an exact model of the selected population, and the inaccuracy of the approximation may be used as a source for exploring new areas.

It has been acknowledged [41] that EDAs build a probability model based on the statistical information extracted from \mathbb{P}_t^S and do sampling from the model to build the next generation. One principle in existing EDAs is to make the probability model approximate the actual probability distribution of the points in \mathbb{P}_t^S as closely as possible within a reasonable computational time. In the case of infinite populations, this principle is equivalent to making \mathbb{P}_{t+1} equal to \mathbb{P}_t^S . Therefore, assuming an infinite population, the behavior of an evolutionary algorithm could be described in this case as

$$\mathbb{P}_{t+1} = \mathbb{P}_t^S. \quad (1)$$

Notice, however, that to guarantee an adequate exploration of new areas of the search space, evolutionary algorithms with finite populations need the equality in Eq. (1) to be relaxed (i.e., \mathbb{P}_{t+1} must be only an approximation of \mathbb{P}_t^S). Because this theoretical analysis focuses on infinite populations, we will use Eq. (1).

The need for feasible methods to calculate \mathbb{P}_t has led to the use of factorizations [21]. A factorized model provides a compact description of the joint probability distribution which can be used to generate new solutions. However, the benefits of using factorized models are achieved at the cost of having just an approximation of \mathbb{P}_{t-1}^S . Therefore, Eq. (1) is not fulfilled in the general case, and the choice of a given factorized model has a number of implications for algorithm performance.

The main feature of UMDA is that it approximates \mathbb{P}_t^S as the product of the univariate frequencies calculated from the selected population. Algorithm 1 shows the UMDA steps (with a finite population).

Algorithm 1. UMDA

```

1  Set  $t \leftarrow 0$ . Generate  $N$  points randomly.
2  do {
3    Select a set  $S$  of  $k \leq N$  points according to a selection method.
4    Compute the univariate marginal frequencies  $p_i^S(x_i, t)$  of  $S$ .
5    Generate  $N$  new points according to the distribution  $p(x, t+1) = \prod_{i=1}^n p_i^S(x_i, t)$ .
6     $t \leftarrow t+1$ .
7  } Until Termination criteria are met.
```

The univariate model assumes that all variables are independent. In such a case, the joint probability \mathbb{P} can be expressed as

$$\mathbb{P}(x) = \prod_{i=1}^n \mathbb{P}(X_i = x_i), \quad (2)$$

where $x = (x_1, x_2, \dots, x_n) \in \Omega$ and X_i denote the i th component of the random vector X . When a probability is factorized according to the univariate model, it is said to be in *Robbin Proportions* (R.p.).

The following equation, originally proposed by Wright [39] for studying population dynamics, will be used in this analysis.

$$\widetilde{W}(p_1, p_2, \dots, p_n) = \sum \prod_{i=1}^n p_i f(x_i), \quad (3)$$

where p_i denotes the univariate probability for the i th gene.

Given the formidable difficulties of modeling finite populations, we consider infinite populations and *proportional selection* for the theoretical analysis. The existence of infinite populations is a strong and widely used assumption [19,20,41].

The *selection probability* associated with a probability \mathbb{P} determined by proportional selection is given by

$$\mathbb{P}^S(x) = \frac{\mathbb{P}(x)f(x)}{\sum_{\tilde{x} \in \Omega} \mathbb{P}(\tilde{x})f(\tilde{x})}. \quad (4)$$

It is known that when proportional selection is used, and Eq. (1) is satisfied for all t , the sequence of probabilities \mathbb{P}_t converges to a probability with mass concentrated at the maxima of f . Zhang and Mühlenbein have proved [41] that this sequence also converges when the tournament and truncation selection schemes are used. However, as mentioned before, the use of the univariate model prevents Eq. (1) from being fulfilled. Actually, with UMDA, \mathbb{P}_{t+1} is only an approximation of \mathbb{P}_t^S .

3. UMDA dynamics for the HardJump function class

The analysis of fitness landscapes in evolutionary algorithms usually imply the choice of a paradigmatic function or class of functions. We follow a similar approach. In this section, we introduce a parametric function that has a constrained number of different values. Thanks to the constraint imposed on the function, we can reduce the complexity of the equations that define the UMDA dynamics and focus on those features of the algorithm that are relevant for this analysis.

3.1. Dynamics for unitation functions

From now on, we will focus on a special class of objective functions. These functions have already been used for the theoretical investigation of evolutionary algorithms [5,18,28,31].

Definition 1. For $x = (x_1, x_2, \dots, x_n) \in \Omega$ we define as the *unitation* of x the expression

$$U(x) := \sum_{i=1}^n x_i.$$

Definition 2. We say that $f : \Omega \rightarrow \mathbb{R}_+$ is a *unitation function* if $f(x) = f(y)$ for all $x, y \in \Omega$ such that $U(x) = U(y)$.

The function U is itself a unitation function, usually called the *Onescounting* or *Onemax* function.

A unitation function achieves a maximum of $n + 1$ different values. Hence, if we consider the sets $U^{-1}(u)$ with $u = 0, 1, 2, \dots, n$, the search for the maxima is constrained to a space of cardinality $n + 1$. We define function $\hat{f} : \{0, 1, \dots, n\} \rightarrow \mathbb{R}_+$, associated with the unitation function f , as

$$\hat{f}(u) = f(x) \quad \text{for } u = U(x).$$

Definition 3. Given a probability \mathbb{P} on Ω , we say that the probability \mathbb{P} is *uniform in unitation class* (u.u.c.) if $\mathbb{P}(x) = \mathbb{P}(y)$ whenever $U(x) = U(y)$.

The cardinality of $U^{-1}(u)$ is $\binom{n}{u}$ for $u \in \{0, 1, \dots, n\}$. Thus, if $U(x) = u$ and \mathbb{P} is u.u.c.,

$$\mathbb{P}(x) = \frac{1}{\binom{n}{u}} \mathbb{P}(U = u). \quad (5)$$

The properties of the sequences of probabilities $\{\mathbb{P}_t\}$ and $\{\mathbb{P}_t^S\}$ follow from the following proposition.

Proposition 1

- (a) If the probability \mathbb{P} is u.u.c., then all univariate marginal probabilities are equal.
- (b) If the probability \mathbb{P} is u.u.c. and f is a unitation function, then the selection probability \mathbb{P}^S (given by Eq. (4)) is u.u.c.
- (c) If the probability \mathbb{P} is in R.p. and all univariate marginal probabilities are equal, then \mathbb{P} is u.u.c. In such a case, the random variable U follows a binomial distribution of parameter p equal to the probability of having a 1 in the first component.
- (d) If \mathbb{P}_0 is u.u.c., then \mathbb{P}_t is u.u.c. for all t .

Proof

(a) : Let $i \in \{1, 2, \dots, n\}$,

$$\mathbb{P}(X_i = 1) = \sum_{u=0}^n \mathbb{P}(X_i = 1, U = u) = \sum_{u=0}^n \sum_{x \in U^{-1}(u): x_i = 1} \mathbb{P}(x) = \sum_{u=0}^n \frac{\mathbb{P}(U = u) \binom{n-1}{u-1}}{\binom{n}{u}} = \frac{1}{n} \sum_{u=0}^n u \mathbb{P}(U = u) \quad (\text{by Equation (5)}). \quad (6)$$

The independence of the last term of the index i confirms the above claim.

(b) Let $x, y \in \Omega$ be such that $U(x) = U(y)$. Because \mathbb{P} is u.u.c., it follows that $\mathbb{P}(x) = \mathbb{P}(y)$. As f is a unitation function, it follows that $f(x) = f(y)$. Therefore,

$$\mathbb{P}^S(x) = \frac{\mathbb{P}(x)f(x)}{\sum_{\tilde{x} \in \Omega} \mathbb{P}(\tilde{x})f(\tilde{x})} = \frac{\mathbb{P}(y)f(y)}{\sum_{\tilde{x} \in \Omega} \mathbb{P}(\tilde{x})f(\tilde{x})} = \mathbb{P}^S(y).$$

(c) Let $x \in \Omega$ and $p = \mathbb{P}(X_1 = 1)$. We now have

$$\begin{aligned} \mathbb{P}(x) &= \prod_{i=1}^n \mathbb{P}(X_i = x_i) \quad (\mathbb{P} \text{ is R.p.}) \\ &= p^{U(x)} (1-p)^{n-U(x)} \quad (\mathbb{P}(X_i = 1) = p \quad \forall i). \end{aligned}$$

Thus, $U(x) = U(y)$ implies $\mathbb{P}(x) = \mathbb{P}(y)$.

Moreover, we have

$$\mathbb{P}(U = k) = \sum_{x \in U^{-1}(k)} \mathbb{P}(x) = \binom{n}{k} p^k (1-p)^{n-k}. \quad (7)$$

(d) Suppose that \mathbb{P}_t is u.u.c. From b) it follows that \mathbb{P}_t^S is u.u.c. Because a) holds, all the univariate distributions of \mathbb{P}_t^S are equal. If we construct \mathbb{P}_{t+1} factorizing with the same univariate distribution of \mathbb{P}_t^S , then by c) we show that \mathbb{P}_{t+1} is u.u.c. This implies the claim. \square

Notice that Part b of [Proposition 1](#) holds for any selection method that uses fitness values only (e.g., proportionate and Boltzmann selection); it would not hold for some specialized selection methods such as those that ensure niching (e.g., restricted tournament selection).

From Proposition 1, and taking \mathbb{P}_0 u.u.c., we construct a sequence $\{\mathbb{P}_t\}$ of probabilities u.u.c. with equal univariate marginal probabilities. Let $p(t)$ be the value $\mathbb{P}_t(X_1 = 1)$. Actually, for any i , $p(t) = \mathbb{P}_t(X_i = 1)$. From the following relations

$$p(t+1) = \mathbb{P}_t^S(X_1 = 1) \quad (\text{by Equation (1)}),$$

$$\mathbb{P}_t^S(X_1 = 1) = \frac{1}{n} \sum_{u=0}^n u \mathbb{P}_t^S(U = u) \quad (\text{by Equation (6)}),$$

$$\mathbb{P}_t^S(U = u) = \frac{\mathbb{P}_t(U = u) \hat{f}(u)}{\sum_{\tilde{u}=0}^n \mathbb{P}_t(U = \tilde{u}) \hat{f}(\tilde{u})} \quad (\text{by Equation (4)}),$$

$$\mathbb{P}_t(U = u) = \binom{n}{u} p(t)^u (1 - p(t))^{n-u} \quad (\text{by Equation (7)}),$$

we have the equation that describes the dynamics of $p(t)$, that is, for $0 < p(t) < 1$

$$p(t+1) = \frac{1/n \sum_{u=0}^n u \binom{n}{u} p(t)^u (1 - p(t))^{n-u} \hat{f}(u)}{\sum_{u=0}^n \binom{n}{u} p(t)^u (1 - p(t))^{n-u} \hat{f}(u)}. \quad (8)$$

The use of proportional selection determines that $p(t) = 0$ implies that $p(t+1) = 0$. However, it holds that function $p(t) \mapsto p(t+1)$ is continuous except at point $p(t) = 0$. Therefore, it is satisfied that

$$\lim_{p(t) \rightarrow 0^+} p(t+1) = \frac{1}{n} \min\{u : \hat{f}(u) > 0\}.$$

In this analysis, we assume that $p(t) = 0$ implies that $p(t+1) = 0$. This is a particular expression of the difference equations for the gene frequencies used by UMDA obtained by Mühlenbein and Mahnig in [18]:

$$p_i(t+1) = p_i(t) + p_i(t) \frac{\frac{\partial W}{\partial p_i}}{W(t)}.$$

Also, we assume that $p(t) = 0$ implies that $p(t+1) = 0$.

3.2. Behavior for the hard jump function class

We begin this section by defining a new class of parametric functions. This function class is a member of the class of unication functions, having a maximal value at $(1, 1, \dots, 1)$ and it is based on *Jump* and related types of functions [18].

Definition 4. The *HardJump*(n, m, δ) function from Ω taking non-negative values with $n \in \mathbb{N}$, $0 < \delta < 1$ and $m \in \{0, 1, \dots, n-1\}$ is defined as

$$\text{HardJump}(n, m, \delta)(x) = \begin{cases} 1, & \text{if } U(x) = n, \\ \delta, & \text{if } U(x) = m, \\ 0, & \text{otherwise,} \end{cases}$$

and the associated function $HJ(n, m, \delta) : \{0, 1, \dots, n\} \mapsto \mathbb{R}_+$ defined as

$$HJ(n, m, \delta)(u) = \begin{cases} 1, & \text{if } u = n, \\ \delta, & \text{if } u = m, \\ 0, & \text{otherwise.} \end{cases}$$

The *HardJump* function class is also related to other functions used for theoretical analysis, such as flat landscapes and functions where there is a unique optimum and all remaining points are constant (needle-in-the-haystack function) [35]. The main feature of the *HardJump* function class is that we can tune both the extension of the plateau and the closeness between suboptimal and optimum values. As we will see later, the ratio $\alpha := m/n \in [0, 1)$ is more important than the parameter m . Taking $m = \alpha n \in \mathbb{N}$, for $HJ(n, \alpha n, \delta)$, Eq. (8) becomes

$$p(t+1) = \alpha + (1 - \alpha) \frac{1}{\delta \binom{n}{\alpha n} \left[\frac{1-p(t)}{p(t)} \right]^{(1-\alpha)n} + 1}. \quad (9)$$

The algorithm convergence analysis can be reduced to just the evolution of $p(t)$. Because the maximum is reached at $(1, 1, \dots, 1)$, the convergence of the algorithm at this point is equivalent to the convergence of $p(t)$ to the value 1.

Let us take a perspective that captures the different facets of the problem. Fig. 1 plots both function $HJ(10, 4, 0.6)$ (circled dots) and the average fitness W (Eq. (3)). Functions $HJ(10, 4, 0.6)$ and W were normalized to be defined in the interval $[0, 1]$. We also present the dynamics $p(t) \mapsto p(t+1)$ analytically expressed by Eq. (9) for the function $HJ(10, 4, 0.6)$.

In general, the plot shape is similar for different values of α , δ and n . Apart from the zero point, there are usually three equilibrium points for $p(t) \mapsto p(t+1)$. As the graph shows, there are two attractors ($p(t) = 1$ and p_a) and a repelling point p_r .

The graph gives a clear indication of where to look for the answers to some of the most important questions of our analysis. If, at the start, $p(0) \in (0, p_r)$, then $p(t)$ will converge monotonically to p_a , which implies convergence to the suboptimum. Conversely, if $p(0) \in (p_r, 1]$, then $p(t)$ will converge to the optimum. Therefore, we can predict from the very start whether the algorithm will converge to the maximum of the function. There are some cases in which there is only one equilibrium point, $p(t) = 1$ (see, for example, the graph for $\alpha = 0.8$, $\delta = 0.8$ and $n = 10$ in Fig. 2). In such cases, the algorithm will always converge. Another possibility is that the points p_r and p_a will coincide at a single point p . In this case, if at the start $p(0) \in (0, p]$, then $p(t)$ will converge to p . Conversely, if $p(0) \in (p, 1]$, then $p(t)$ will converge to 1.

In what follows, we will study the transformations observed in the dynamics of $p(t)$ for different values of α , δ and n . This analysis can be translated to the study of the equilibrium points p_a and p_r .

Fig. 2 shows that W is a smoother version of the original *HardJump* function, which agrees with Mühlenbein and Mahnig's theoretical analysis [18] for other functions. Taking a closer look, we can see how the dynamics of $p(t)$ are related to W . Those values of $p(t)$ that do not converge to the optimum are to be found in the graph in the basin of attraction of this suboptimum in W . The dynamics of $p(t)$ clearly reflect the algorithm's convergence conditions. It also gives an idea of its convergence rate. This information cannot be extracted from W .

In Fig. 2 (left) we show the $p(t) \mapsto p(t+1)$ dynamics for the *HardJump* $(10, 10\alpha, 0.8)$ function for different values of α . For $\alpha = 0$ the two separate optima in W are the farthest apart. As α is increased, the local optimum shifts to the right. For $\alpha = 0.8$, the local and global optima basins of attraction are very close. In this case, we expect the UMDA to be able to jump to the global optimum. The analysis of the dynamics of $p(t)$ supports the hypothesis we propose regarding UMDA: for small values of α , the algorithm is quite likely to get trapped in a local attractor. As α is increased, the algorithm is more likely to converge to the optimum.

Fig. 2 (right) shows the $p(t) \mapsto p(t+1)$ dynamics for different values of δ ($\alpha = 0.4$ and $n = 10$). In this case, we can analyze the effect of making the values of the local and global optima closer. When δ is increased, the value of the local optimum increases. The dynamics graph shows the influence of this change on UMDA behavior. The higher the local optimum is, the wider the range will be of probabilities for which the algorithm will get trapped in a local attractor.

Fig. 9 shows the dynamics of $p(t) \mapsto p(t+1)$ for different values of n ($\alpha = 0.4$ and $\delta = 0.5$). This figure suggests that $p(t) \mapsto p(t+1)$ behaves asymptotically when the dimension of the search space grows to infinity. In Section 6, we analytically show that this limit behavior does exist.

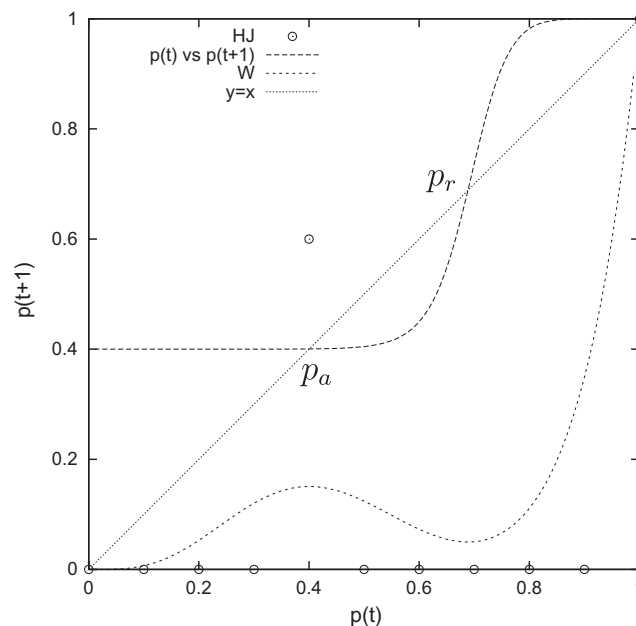


Fig. 1. Functions: W (normalized), $p(t) \mapsto p(t+1)$ and $HJ(10, 4, 0.6)$ (normalized). Equilibrium points p_a (attractor) and p_r (repeller).

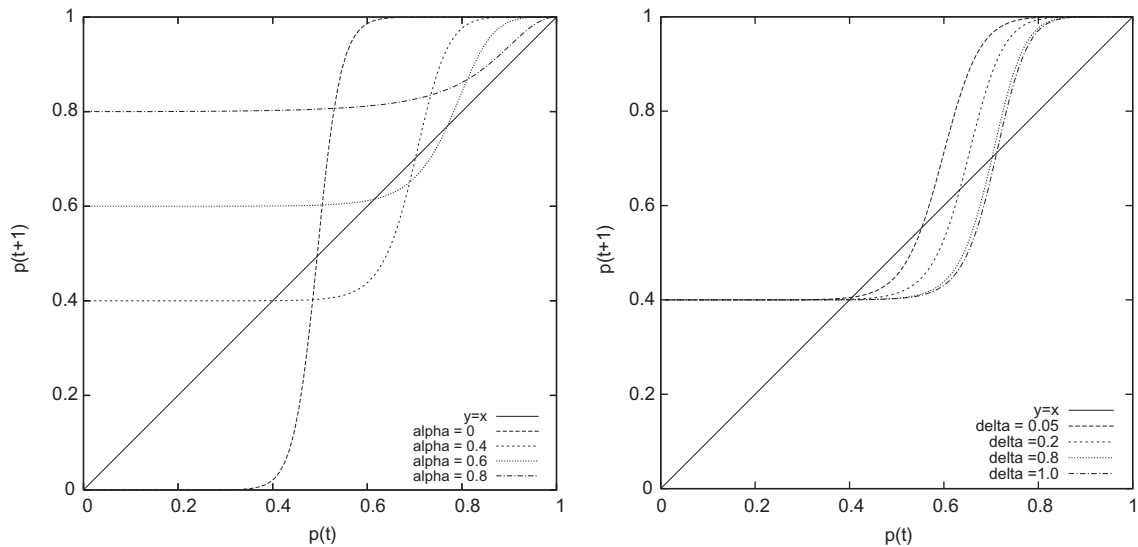


Fig. 2. Graph of $p(t)$ versus $p(t+1)$ for $\text{HardJump}(10, 10\alpha, \delta)$. Left: $\delta = 0.8$ and $\alpha = 0, 0.4, 0.6, 0.8$. Right: $\alpha = 0.4$ and $\delta = 0.05, 0.2, 0.8, 1$.

4. Limits of the infinite population approximation

We start by explaining the implications of considering a finite rather than infinite population. Our study is based on the analysis of function $p(t) \mapsto p(t+1)$. The main property that makes this analysis valid is that all the univariate marginal probabilities are equal under the hypothesis of Proposition 1. When considering a finite population S , this hypothesis no longer holds; in particular, Eq. (3) becomes:

$$\mathbb{P}^S(x) = \frac{\mathbb{P}^E(x)f(x)}{\sum_{y \in S} \mathbb{P}^E(y)f(y)}, \quad (10)$$

where \mathbb{P}^E represents only an approximation of the probability distribution \mathbb{P} . \mathbb{P}^E is the empirical distribution (we use super-index E to represent this fact) computed from the sample S , i.e.,

$$\mathbb{P}^E(x) = \frac{1}{|S|} \sum_{y \in S} \mathbf{1}_{(y=x)}.$$

The main problem with this change is that \mathbb{P}^E is a random distribution; therefore, \mathbb{P}^S will also be a random distribution. The second and most difficult problem to solve is that the random distribution \mathbb{P}^E is not u.u.c. in the general case and it no longer has the property of all the marginal distributions being equal. This means that the algorithm dynamics analysis using function $p(t) \rightarrow p(t+1)$ needs to be restated.

We simulated UMDA a number of times using finite selected populations. Let us describe the results of the experiments and analyze their implications.

Given a function $HJ(n, m, \delta)$ and a predefined population size N , we run r experiments with UMDA, taking \mathbb{P}_0 as an initial probability, where \mathbb{P}_0 is the probability distribution resulting from the product of equal univariate marginal distributions equal to $p(0) = i/r$, $i = 1, 2, \dots, r$. The bar charts in Fig. 3 show the results of the experiments.

In each chart, the bars represent the number of experiments for values of $p(0)$ stated at the base of the bar. The area of the bar shaded in white (light grey and dark grey, respectively) represents the total proportion of experiments that converged to the value 0 (δ and 1, respectively). For clarity's sake, the function $p(t) \rightarrow p(t+1)$ has been represented as a solid line and the diagonal as a dashed line.

Fig. 3 (a) represents the results for $HJ(10, 4, 0.6)$ with a population size $N = 10^4$ and with $r = 10^3$. In this case, the width of the bars is 0.01. Therefore, each bar represents 10 runs.

This experiment serves to show the accuracy of the approximation given by the theoretical result we obtained, even if the u.u.c. conditions do not hold. The chart shows that, on average, the behavior confirms our theoretical assessment. However, the behavior in the neighborhood of point $(0, 0, \dots, 0)$ departs from the previous theoretical results. We will analyze this particular behavior later.

Fig. 3 (b) shows the results for $HJ(10, 4, 0.6)$, with a population size $N = 100$ and with $r = 200$. The width of the bars is $1/3$ in both this and the following figures. Notice how the theoretical results are a good approximation of UMDA's real behavior. Fig. 3 (c) shows the results for $HJ(20, 8, 0.6)$, with a population size $N = 1000$ and $r = 200$. The size of the selected population is 0.1% of the total population. Even so, the results achieved are quite good. Fig. 3 (d) shows the results for $HJ(20, 8, 0.6)$,

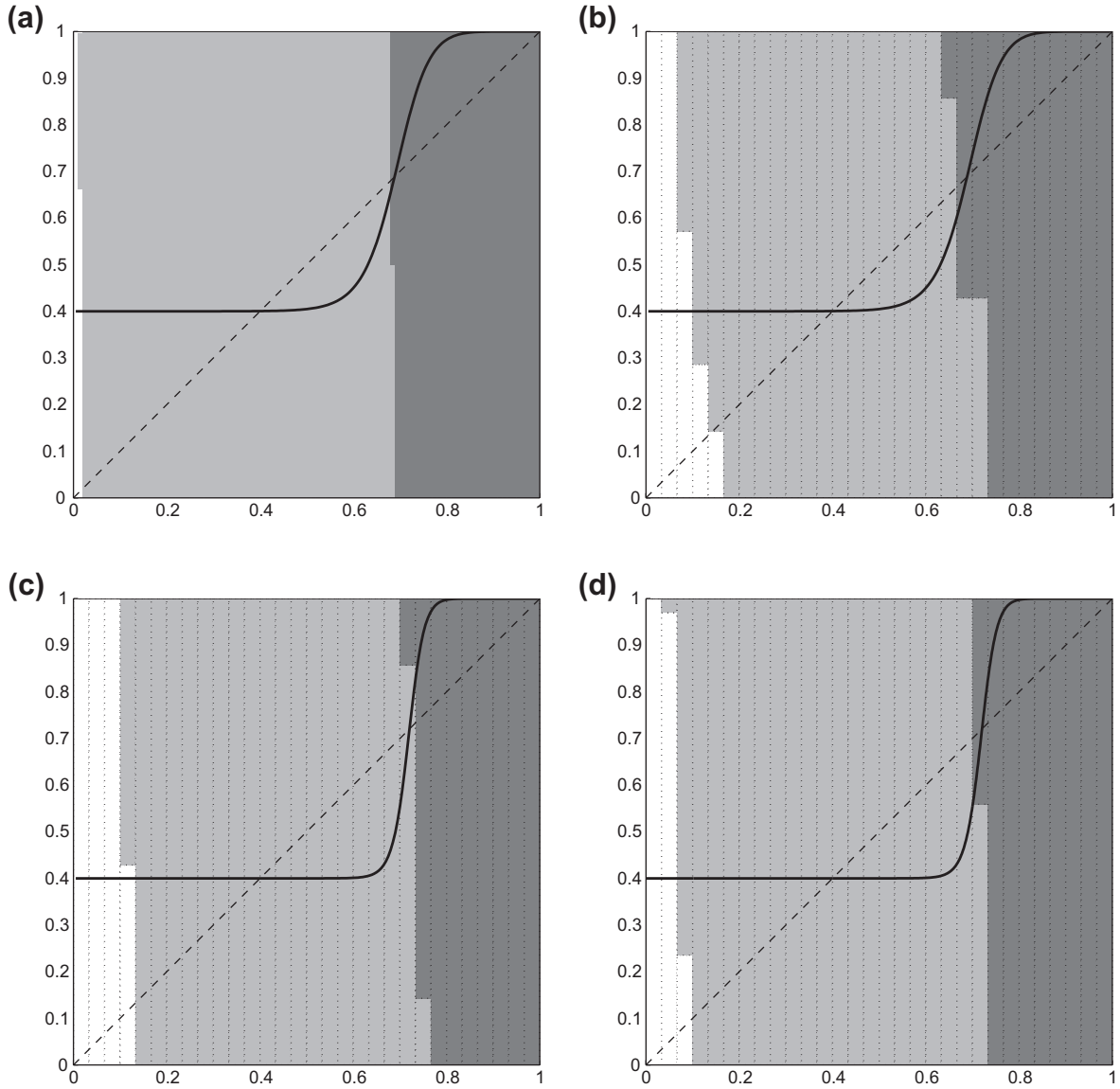


Fig. 3. Bar charts representing the proportion of runs converging to the UMDA attractors for finite populations of different sizes: (a) Function $HJ(10,4,0.6)$, $N = 10000$, $r = 200$. (b) Function $HJ(10,4,0.6)$, $N = 100$, $r = 200$. (c) Function $HJ(20,8,0.6)$, $N = 1000$, $r = 200$. (d) Function $HJ(20,8,0.6)$, $N = 10000$, $r = 200$.

where the population size was $N = 10^4$ and $r = 200$. Notice that, in this case, the behavior of the algorithm in the neighborhood of point $(0,0,\dots,0)$ is closer to the assessments we made, which was to be expected when N is increased.

4.1. Behavior in the neighborhood of point $(0,0,\dots,0)$

The observed behavior of UMDA with a finite population in the neighborhood of point $(0,0,\dots,0)$ can be explained by $(0,0,\dots,0)$ being a fixed point of $p(t) \rightarrow p(t+1)$. When we consider $N = \infty$, this point has a probability of 0, but when the population size is finite, the probability of picking up $(0,0,\dots,0)$ is positive (unless the probability is concentrated at $(1,1,\dots,1)$). Consequently, it begins to act as an attractor, and it is able to make the system converge to it. This is the case for points located within a small neighborhood of $(0,0,\dots,0)$. Furthermore, when $N \ll 2^n$, the UMDA behavior becomes more erratic because each selected point leads to a higher probability. Hence, the behavior can be unlike the theoretical predictions we made (see Fig. 4).

To explain this phenomenon in detail, let us analyze Eq. (10). Specifically, let us analyze what happens to p_1^S , which is the first univariate marginal probability \mathbb{P}^S . It turns out that $p_1^S > 0$ if and only if there exists some $x \in S$ such that $f(x) > 0$, (i.e., $U(x) \in \{m, 1\}$) and its first component x_1 is 1. For the sake of simplicity, let us suppose now that all the univariate marginals

are equal to p . It holds, then, that the probability of selecting an x such that $U(x) \in \{m, 1\}$ is $\binom{n}{m} p^m (1-p)^{n-m} + p^n$. Thus, we obtain the following probability

$$\mathbb{P}(\{p_1^S > 0\}) = 1 - \left[1 - \binom{n}{m} p^m (1-p)^{n-m} - p^n \right]^N. \quad (11)$$

It is not difficult to see that when p is very small, the probability p_1^S drastically decreases (like all the univariate marginal probabilities); therefore, the algorithm will approach point $(0, 0, \dots, 0)$. Not so obvious is that $\binom{n}{m} p^m (1-p)^{n-m} + p^n$ reaches its local extrema when $p \approx m/n$ and p^n for $p = 1$ (see solid line in Fig. 5). Therefore, if N is very small with respect to 2^n , the probability $\mathbb{P}(\{p_1^S > 0\})$ will not be close to 1. Thus, there is a high probability that the algorithm will be attracted by $(0, 0, \dots, 0)$. Fig. 5 shows the graph representing $\mathbb{P}(\{p_1^S > 0\})$ against p , for different values of N . The case $N = 100$ is particularly interesting. In this set up, for $p \approx 0.73$, the algorithm has a higher probability of converging to zero than to any other point.

5. Beyond the HardJump functions

So far, this paper has studied UMDA, analyzing the dynamics of the univariate probabilities from a theoretical point of view. We have been able to obtain explicit expressions of UMDA dynamics and show that they match empirical experiments and simulations.

We have shown that, as with other unification functions, the HardJump function has a number of properties that makes it suitable for an explicit analysis and theoretical calculus. It has also been proved that its analysis is useful for understanding the dependence between the behavior of UMDA (and other EDAs) and the distance between the optimum and the closer sub-optima of the function. This means that the function parameters can be tuned to analyze the algorithm's capacity to escape from local suboptima.

However, the class of HardJump functions analyzed in previous sections has only two non-zero fitness values. Furthermore, we have focused on the case of proportional selection. In functions of practical interest, the number of non-zero fitness values is expected to be higher. Practical implementations of UMDA and other EDAs may use other selection schedules.

In this section, we will show how results proved for HardJump can be applied to functions with more than two non-zero fitness values and how they can be useful for analyzing other selection schemes.

5.1. On the neighborhood of $(1, 1, \dots, 1)$

We start by studying the dynamics in a neighborhood of the point $(1, 1, \dots, 1)$. Let

$$u_0 = \max\{0 \leq u < n : \hat{f}(u) > 0\}.$$

We denote by Q the mapping $p = p(t) \rightarrow p(t+1) = Q(p)$.

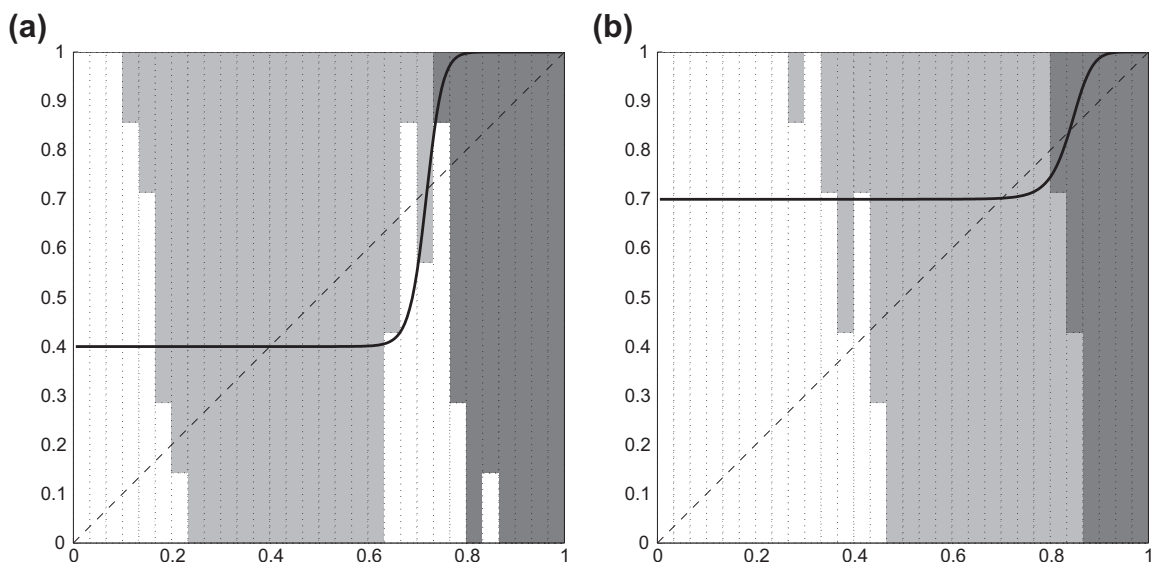


Fig. 4. Bar chart of the UMDA attractors with finite populations. (a) Function $HJ(20, 8, 0.6)$, $N = 100$, $r = 200$. (b) Function $HJ(20, 14, 0.6)$, $N = 100$, $r = 200$.

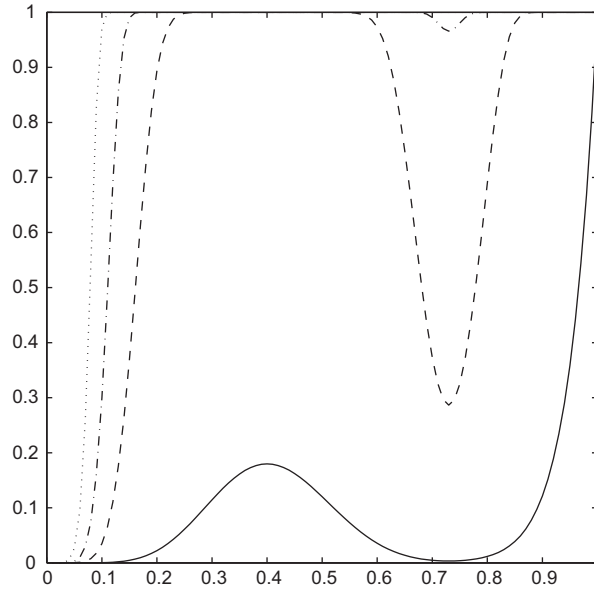


Fig. 5. $\mathbb{P}(\{p_i^s > 0\})$ against p (see Eq. (11)) when $N = 1$ (solid line), when $N = 100$ (dashed line), when $N = 1,000$ (dash-dot line) and when $N = 10,000$ (dotted line). In all cases, $n = 20$ and $m = 8$.

We now have the following

$$\begin{aligned} Q(p) &= \frac{\sum_{u=0}^n u \binom{n}{u} p^u (1-p)^{n-u} \hat{f}(u)}{n \sum_{u=0}^n \binom{n}{u} p^u (1-p)^{n-u} \hat{f}(u)} = \frac{n + u_0 \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0) + (1-p)^{n-u_0+1} \sum_{u=0}^{u_0-1} u \binom{n}{u} p^u (1-p)^{u_0-u-1} \hat{f}(u)}{n + n \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0) + (1-p)^{n-u_0+1} \sum_{u=0}^{u_0-1} \binom{n}{u} p^u (1-p)^{u_0-u-1} \hat{f}(u)} \\ &= \frac{n + u_0 \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0) + o((1-p)^{n-u_0})}{n + n \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0) + o((1-p)^{n-u_0})} = \frac{n + u_0 \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0)}{n + n \binom{n}{u_0} p^{u_0} (1-p)^{n-u_0} \hat{f}(u_0)} + o((1-p)^{n-u_0}) \\ &= \hat{Q}(p) + o((1-p)^{n-u_0}) \end{aligned}$$

where \hat{Q} is the mapping representing $p(t) \rightarrow p(t+1)$ when the objective function is $HJ(n, u_0, \hat{f}(u_0))$. Consequently, it holds that for values of p close to 1, the dynamics related to function \hat{f} are similar to that of function $HJ(n, u_0, \hat{f}(u_0))$. Obviously, this approximation will improve the lower u_0 is, i.e., as the gap increases. It is not so clear that the approximation is better when the $\sup\{m : \hat{f}(u) = 0, \forall u \in \{m+1, m+2, \dots, u_0-1\}\}$ decreases.

Therefore, when the convergence valley of the optimum is within the vicinity of $(1, 1, \dots, 1)$, it is well approximated by the convergence valley for function $HJ(n, u_0, \hat{f}(u_0))$. Conversely, when the convergence valley of the optimum is larger than the neighborhood with a valid approximation, nothing can be said about it.

Let us look at this graphically. Fig. 6 (a) represents the dynamics of the functions

$$\tilde{f}(m) = \begin{cases} 1/4, & m = 5, \\ 1/2, & m = 10, \\ 3/4, & m = 15, \\ 1, & m = 20, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

with $n = 20$ and $HJ(20, 15, 3/4)$. Clearly, this approximation is sufficient to describe the convergence valley. Note that, in this case, the closest sub-optimum competes against the optimum. Fig. 6 (b) represents the opposite case.

The graph in Fig. 6 (b) represents the dynamics for functions

$$\hat{f}(m) = \begin{cases} 0.8, & m = 8, \\ 0.8, & m = 18, \\ 1, & m = 20, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

with $n = 20$ and $HJ(20, 15, 3/4)$.

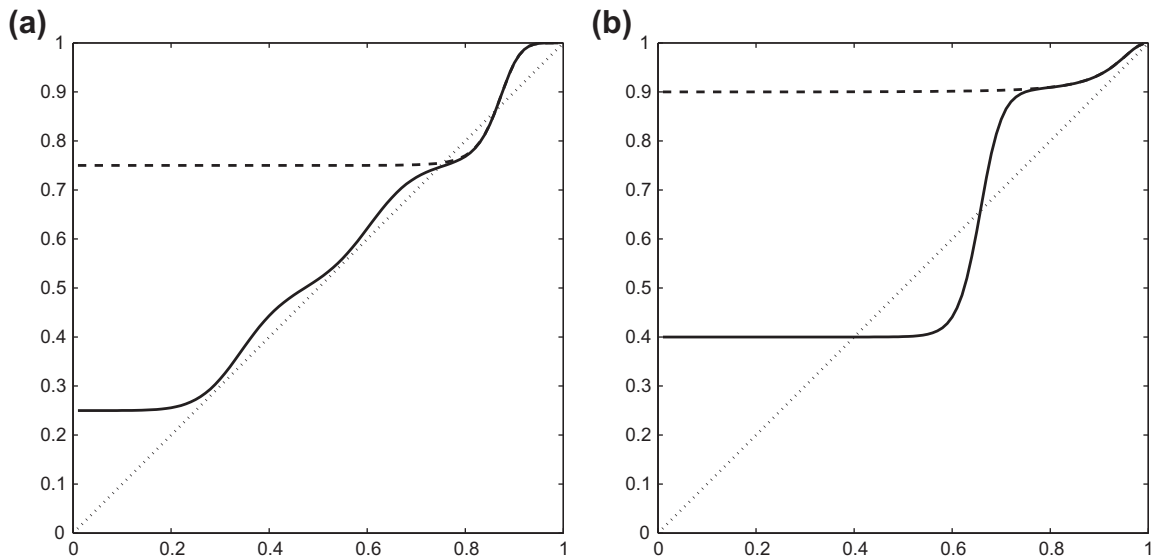


Fig. 6. Graph of $p(t)$ against $p(t+1)$ for the functions (a) \tilde{f} , (see Eq. (12)) (solid line) and $HJ(20, 15, 3/4)$ (dashed line); (b) \tilde{f} , (see Eq. (13)) (solid line) and $HJ(20, 18, 0.8)$ (dashed line).

Now, the closest suboptimum does not compete with the optimum because it is located too close to it. It is actually helpful to make the convergence valley of the optimum larger. Therefore, the convergence valley does not support a valid approximation.

In these cases, we have found (data not shown) that it is possible to approximate the convergence valley by studying the dynamics associated with the new function $\hat{g} : \{0, 1, \dots, u_0\} \rightarrow \mathbb{R}$, defined as

$$\hat{g}(x) = \frac{\hat{f}(m)}{\hat{f}(u_0)}.$$

In Fig. 7, we represent the dynamics of the original function \tilde{f} in Eq. (13) and the dynamics of $\hat{f} = HJ(18, 7, 1)$, scaled to be in the square $\{(0, 0), (0, 0.9), (0.9, 0), (0.9, 0.9)\}$. Because the closest suboptimum (located at $m = 18$) does not compete against the

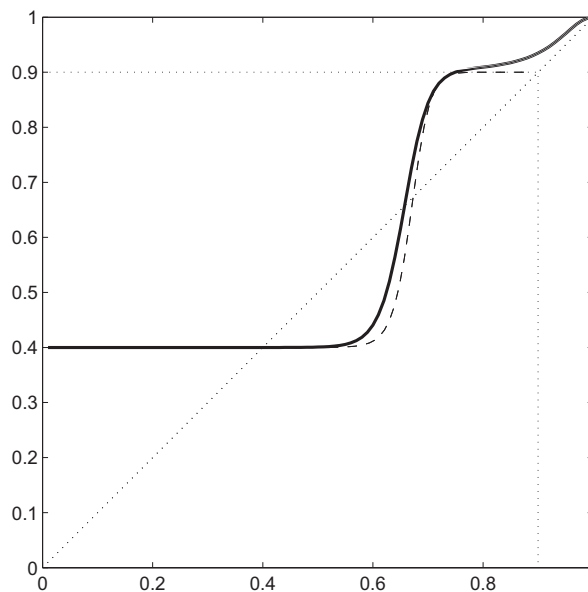


Fig. 7. Graph of $p(t)$ against $p(t+1)$ for \tilde{f} (Eq. (13)) (solid line), $\hat{g} = HJ(18, 7, 1)$ (dashed line) (scaled to square with vertex $\{(0, 0), (0, 0.9), (0.9, 0), (0.9, 0.9)\}$). The square and the diagonal are represented by dotted lines.

optimum ($m = 20$), the convergence valley of \hat{g} would appear to be able to be approximated using the convergence valley for a function with $m = 181$ as its optimum and $m = 8$ as its closest suboptimum. The analysis can be carried out by first scaling and then studying the dynamics of $HJ(18, 7, 1)$ (note that $8/20 \approx 7/18$). The same analytical procedure could be applied to a study of the dynamics of other unitation functions.

5.2. Trap functions and Boltzmann selection

The analysis focused on the *HardJump* class of functions, which contains a sizeable number of zeros. We can extend this analysis to other classes of functions and other types of selection methods. One example is Boltzmann selection.

We recall that under Boltzmann selection, the selection probability is

$$\mathbb{P}^S(x) = \frac{\mathbb{P}(x) \exp(\lambda f(x))}{\sum_{\tilde{x} \in \Omega} \mathbb{P}(\tilde{x}) \exp(\lambda f(\tilde{x}))}, \quad (14)$$

where $\lambda > 0$. In other words, it is equivalent to considering proportional selection (4) with function $\exp(\lambda f(x))$ instead of $f(x)$. From the remark after Proposition 1 we can conclude that the dynamics for Boltzmann selection can be analyzed in a way similar to that for proportional selection. To illustrate this point, we employ an analysis similar to the one previously presented to study a more “common” class of functions. We work with a *trap* function defined as

$$f(x) = \begin{cases} \frac{a}{m}(m - U(x)), & \text{if } U(x) \leq m, \\ \frac{b}{n-m}(U(x) - m), & \text{otherwise.} \end{cases}$$

This function has two local extrema in $x = (0, 0, \dots, 0)$ and $(1, 1, \dots, 1)$. We want to compare the dynamics of this function with Boltzmann selection to the dynamics of $HJ(10, 0, \exp(\lambda(f(x) - 1)))$ with proportional selection.

Let us consider the *trap* function with parameters $a = 0.4$, $b = 1$, $n = 10$ and $m = 4$ (Fig. 8 left). Fig. 8 (right) plots the dynamics for Boltzmann selection with $\lambda = 25$ and the dynamics of the corresponding *HardJump* function with proportional selection.

The similarity of the dynamics of both functions encourages us to think that, under proportional selection, the dynamics for the *HardJump* class function is a good representation of the dynamics for functions in which the optimum and suboptimum are much larger than the other values (not necessarily equal to zero). This is the case, for example, when using Boltzmann selection with a high λ value.

6. Long string limit analysis

Long string limit analysis refers to the theoretical analysis of some evolutionary algorithms' quantities when the number of variables (length of the string) tends toward infinity. This type of analysis has been conducted within the framework of statistical mechanics to study the performance of GAs [26]. It eliminates some of the details hidden in the generic behavior of evolving populations. Additionally, the limit can give an accurate qualitative description of a GA for moderate-sized strings. One of the drawbacks of this type of study for GAs is that the inclusion of different genetic operators complicates the analysis. The finite string limit is unobtainable for most problems [26].

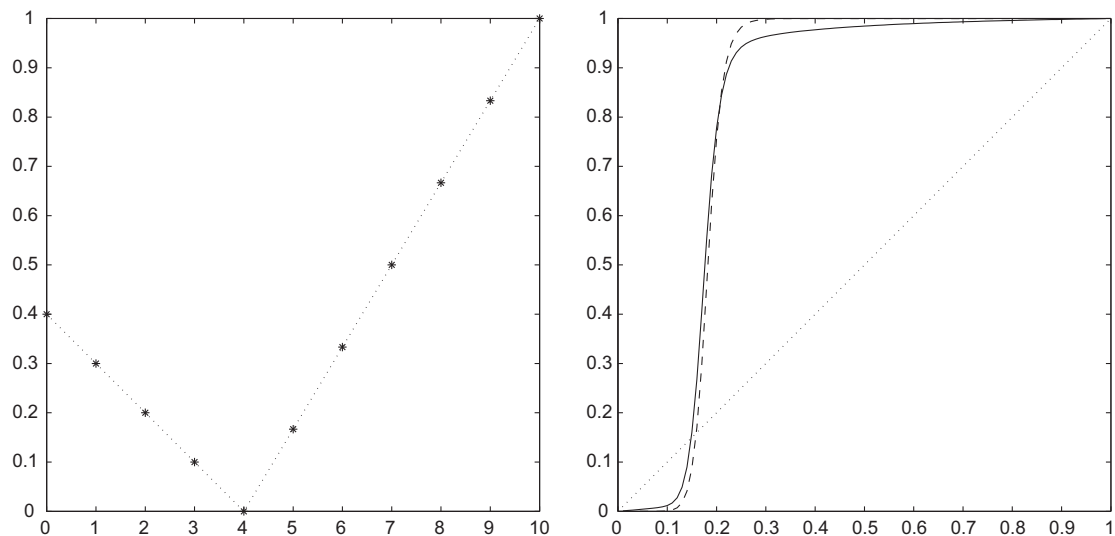


Fig. 8. (Left) A trap function and (right) the dynamics of a trap function using Boltzmann selection ($\lambda = 25$) and the corresponding *HardJump* function using proportional selection (dashed line).

In this section, we present a long string analysis of UMDA dynamics. We prove the existence of a limit behavior of $p(t) \mapsto p(t+1)$ when $n \mapsto \infty$ and, using simulations, we investigate how this limit is related to the behavior of the theoretical UMDA for moderate-sized problems. An important assumption for this analysis is that we consider $p(0)$ to be fixed. Only n is increased. This means that we do not consider the $1/n$ effects on $p(0)$ which is a fundamental difference from the long string limit analysis of the GAs presented in [26], where this effect is considered.

Fig. 9 shows that the long string limit behavior of $p(t) \mapsto p(t+1)$ presents a highly accurate prediction of the behavior of UMDA for moderate-sized problems: $n \geq 50$.

Proposition 2. For all instants t

$$\lim_{n \rightarrow \infty} p(t+1) = \begin{cases} 1, & \text{if } p(t) > p_r^\alpha, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$p_r^\alpha = [1 + \alpha^{1-\alpha}(1-\alpha)]^{-1}.$$

Proof. Using Stirling's formula $n! = \sqrt{2\pi n} n^{n+1/2} e^{-n} (1 + \epsilon_n)$ with $\epsilon_n \rightarrow 0$ when $n \rightarrow \infty$ we have

$$\binom{n}{\alpha n} = (2\pi\alpha(1-\alpha)n)^{-1/2} [\alpha^\alpha (1-\alpha)^{1-\alpha}]^{-n} (1 + \epsilon_{\alpha,n}),$$

where $\alpha \in [0,1]$ such that $\alpha n \in \mathbb{N}$ and $\epsilon_{\alpha,n} \rightarrow 0$ when $n \rightarrow \infty$.

Thus, we can write Eq. (9), for $p(t) \in]0,1[$, as

$$p(t+1) = \alpha + (1-\alpha) \left[1 + \frac{\kappa_n}{\sqrt{n}} \left(\frac{1-p(t)}{p(t)} \right)^{(1-\alpha)n} \frac{1}{\alpha^{2n}(1-\alpha)^{(1-\alpha)n}} \right],$$

where $\kappa_n \rightarrow 1/\sqrt{2\pi\alpha(1-\alpha)}$ as $n \rightarrow \infty$. It follows that $p(t+1) \rightarrow 1$ when

$$\left(\frac{1-p(t)}{p(t)} \right)^{(1-\alpha)n} \frac{1}{\alpha^{2n}(1-\alpha)^{(1-\alpha)n}} > 1, \quad (15)$$

otherwise $p(t+1) \rightarrow \alpha$. The relation (15) is equivalent to

$$p(t) < \frac{1}{1 + \alpha^{2/(1-\alpha)}(1-\alpha)} = p_r^\alpha. \quad \square$$

As Fig. 9 shows, for all n the algorithm will tend toward the optimum of the function whenever $p > p_r^\alpha$.

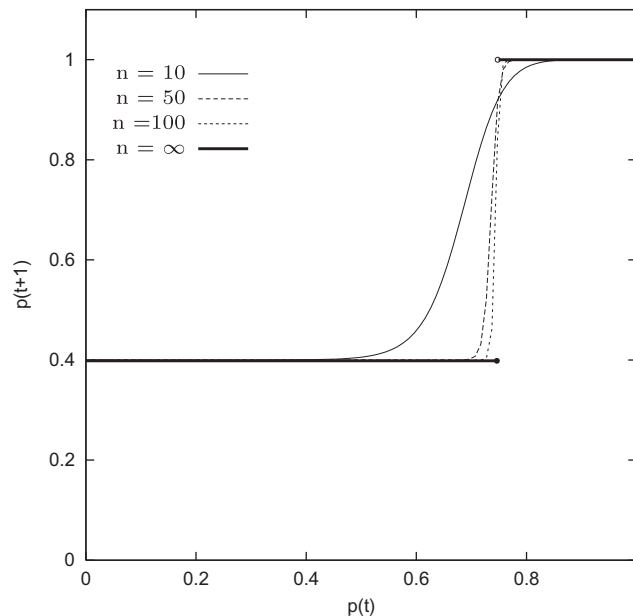


Fig. 9. $p(t)$ against $p(t+1)$ for $\text{HardJump}(n,4,0.8)$ with $n = 10, 50, 100, \infty$.

7. Related problems and applications

HardJump is a function class in which the function is defined only in terms of the unitation constraint imposed by the parameters. The modeling of the dynamics presented for *HardJump* serves to analyze the impact of the unitation constraint on the algorithm dynamics. This can be done by considering the influence parameters m and δ have on algorithm performance.

These parameters can be even more useful for studying the influence of unitation constraints on the generation of dependencies during the search and, therefore, for investigating the performance of EDAs that learn the probabilistic model structure. Initial steps in this direction were presented in [32,33], in which the strength of pairwise dependencies for functions with different unitation constraints were scrutinized, and sampling operators that take these constraints into consideration were proposed. Distinguishing constraint-induced dependencies from dependencies determined by other characteristics of the function structure can be crucial for designing efficient sampling techniques in EDAs and creating repair operators that satisfy the key problem dependencies as closely as possible. The *HardJump* function (and its possible extensions) explicitly punishes all solutions that violate a given unitation constraint, and tuning the rest of the parameters is a natural way to study the generation of dependencies.

Although the results presented in this paper are mainly of theoretical value, we briefly analyze a number of problems to which these results could be applied.

Functions with unitation constraints [32] are functions in which the search is limited to solutions that have a constrained range of unitation values. There are several problems that can be approached as the optimization of a function with unitation constraints. Examples include the search of graph substructures [29], feature subset selection constraining the size of the feature subsets [27], and QSAR studies in chemistry [1]. This type of function has also been used to investigate the effects of adding general constraints on the generation of probabilistic dependencies in EDAs [33].

In [19], Mühlenbein and Mahnig identify the systems dynamics approach to optimization, and they analyze its relationship to EDAs. The systems dynamics approach works by directly evolving the probability distributions without having to generate populations of solutions. The evolution is guided by dynamic difference equations. The difference equation that describes the dynamics of $p(t)$ given in Section 3.1 could be used to design this kind of algorithm to solve unitation functions.

8. Conclusions

Mühlenbein and Mahnig [19] noted that if there is a tendency to the optimum in the average fitness W , UMDA will be able to find it. We have found that, in the case of Jump functions, the existence of such a tendency will depend on the aspects of difficulty encoded in the function parameters that are also common to other optimization algorithms. The parameter α controls the number of local optima of the function as well as the gap between these optima and the global optimum point. The parameter δ expresses the ratio between the local optima and the single global optimum value.

An analysis of the equations that describe UMDA behavior for *HardJump* revealed that the algorithm convergence is much more sensitive to a change in parameter α than to a change in δ . This can be illustrated by the following paradox. Let us suppose that UMDA is able to converge to the global optimum when $\delta < 1$. If we increase δ to just over 1, then the algorithm will still have a high probability of reaching the previous global (now local) optimum. This deceptive behavior explains how sensitive the algorithm is to the existence of many local optima. This issue deserves further consideration, which we have left for future work.

We have conducted extensive simulations to investigate the implications of using finite populations, and we have shown the usefulness of these assessments in this case. Additionally, we have explained how the theoretical analysis can be extended to functions with more than two zero-values.

The study of stochastic search algorithm dynamics and the analysis of long string limit has been recognized as a current trend in theoretical computer science. In this paper, we have presented a long string analysis of the dynamics of UMDA for optimizing a difficult class of parametric functions. The long string limit analysis achieved three things. First, it demonstrated the existence of a limit p_r^z for the repelling equilibrium point when $n \rightarrow \infty$. This is the first analysis of its kind published in the field of EDAs. Second, the analysis derived sufficient convergence conditions for the algorithm based on p_r^z . Third, it described the algorithm's behavior for moderate-sized problems obtained from the limit.

We expect work on the analysis of theoretical functions such as *HardJump* to provide many more answers to some of the questions that arise from the use of EDAs. One example is the relationship between the type of function difficulty and the algorithm behavior. We also anticipate that the study of EDA long string limit behavior will become part of the regular theoretical analysis of these algorithms.

Acknowledgements

The authors thank the reviewers for useful comments on the paper. Part of this work was conducted during Roberto Santana's stay at the Institute of Cybernetics, Mathematics, and Physics (ICIMAF), Havana, Cuba. This work has been partially supported by the Ertortek, Saiotek and Research Groups 2007–2012 (IT-242-07) programs (Basque Government), the CajalBlueBrain project, TIN2008-06815-C02-01, TIN-2010-20900-C04-04 and Consolider Ingenio 2010 – CSD2007-00018

projects (Spanish Ministry of Science and Innovation) as well as the COMBIOMED network in computational biomedicine (Carlos III Health Institute).

References

- [1] Y.M. Alvarez-Ginarte, R. Crespo, L.A. Montero-Cabrera, J.A. Ruiz-Garcia, Y.M. Ponce, R. Santana, E. Pardillo-Fontdevila, E. Alonso-Becerra, A novel in-silico approach for QSAR studies of anabolic and androgenic activities in the 17-hydroxy-5-androstane steroid family, *QSAR & Combinatorial Science* 24 (2005) 218–226.
- [2] S. Baluja, Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [3] S. Baluja, R. Caruana, Removing the genetics from the standard genetic algorithm, Technical Report CMU-CS-95-141, Carnegie-Mellon University, Pittsburgh, 1995.
- [4] E. Cantú-Paz, Feature subset selection with hybrids of filters and evolutionary algorithms, in: M. Pelikan, K. Sastry, E. Cantú-Paz (Eds.), *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Studies in Computational Intelligence, Springer-Verlag, 2006, pp. 291–314.
- [5] U.K. Chakraborty, H. Mühlenbein, Linkage equilibrium and genetic algorithms, *Proceedings of the 1997 Congress on Evolutionary Computation CEC-1997*, vol. 2, IEEE press, 1997, pp. 25–29.
- [6] D.E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academics Publishers, 2002.
- [7] C. González, J.A. Lozano, P. Larrañaga, Analyzing the PBIL algorithm by means of discrete dynamical systems, *Complex Systems* 12 (4) (2001) 465–479.
- [8] C. González, J.A. Lozano, P. Larrañaga, Mathematical modeling of discrete estimation of distribution algorithms, in: P. Larrañaga, J.A. Lozano (Eds.), *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002, pp. 143–162.
- [9] J. Gahl, P.A. Bosman, S. Minner, Convergence phases, variance trajectories, and runtime analysis of continuous edas on the sphere function, in: D. Thierens et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2007*, vol. I, ACM Press, New York, NY, USA, 2007, pp. 516–522.
- [10] L. Grosset, R. Riche, R.T. Haftka, A double-distribution statistical algorithm for composite laminate optimization, *Structural and Multidisciplinary Optimization* 31 (1) (2006) 49–59.
- [11] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation* 3 (4) (1999) 287–297.
- [12] P. Larrañaga, J.A. Lozano (Eds.), *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.
- [13] J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (Eds.), *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, Springer-Verlag, 2006.
- [14] T. Mahnig, *Populations basierte Optimierung durch das Lernen von Interaktionen mit Bayes'schen Netzen*, PhD thesis, University of Bonn, Sankt Augustin, Germany, 2001, GMD Research Series No. 3/2001.
- [15] H. Mühlenbein, The equation for response to selection and its use for prediction, *Evolutionary Computation* 5 (3) (1997) 303–346.
- [16] H. Mühlenbein, U.K. Chakraborty, Gene pool recombination, genetic algorithm and the onemax, *Journal of Computing and Information Technology* 5 (3) (1997) 167–182.
- [17] H. Mühlenbein, T. Mahnig, Convergence theory and applications of the Factorized distribution algorithm, *Journal of Computing and Information Technology* 7 (1) (1998) 19–32.
- [18] H. Mühlenbein, T. Mahnig, *Theoretical Aspects of Evolutionary Computing*, Chapter Evolutionary Algorithms: From Recombination to Search Distributions, Springer Verlag, Berlin, 2000, pp. 137–176.
- [19] H. Mühlenbein, T. Mahnig, Evolutionary computation and beyond, in: Y. Uesaka, P. Kanerva, H. Asoh (eds.), *Foundations of Real-World Intelligence*, CSLI Publications, Stanford, California, 2001, pp. 123–188.
- [20] H. Mühlenbein, T. Mahnig, Evolutionary algorithms and the Boltzmann distribution, in: K.A. DeJong, R. Poli, J. Rowe (eds.), *Foundation of Genetic Algorithms 7*, Morgan Kaufmann, 2002, pp. 133–150.
- [21] H. Mühlenbein, T. Mahnig, A. Ochoa, Schemata, distributions and graphical models in evolutionary optimization, *Journal of Heuristics* 5 (2) (1999) 213–247.
- [22] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature - PPSN IV*, LNCS, vol. 1141, Springer Verlag, Berlin, 1996, pp. 178–187.
- [23] M. Pelikan, Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms. *Studies in Fuzziness and Soft Computing*, Springer, 2005.
- [24] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, Bayesian optimization algorithm, population sizing, and time to convergence, in: *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2000*, 2000, pp. 275–282.
- [25] M. Pelikan, D.E. Goldberg, F. Lobo, A survey of optimization by building and using probabilistic models, *Computational Optimization and Applications* 21 (1) (2002) 5–20.
- [26] A. Prügel-Bennet, On the long string limit, in: W. Banzhaf, C. Reeves (eds.), *Proceedings of Foundations of Genetic Algorithms 5*, 1999, pp. 45–56.
- [27] Y. Saeyns, S. Degroove, D. Aeyels, Y. Van de Peer, P. Rouzé, Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction, *Bioinformatics* 19 (2) (2003) ii179–ii188.
- [28] R. Santana, Study of neighborhood search operators for unitation functions, in: *Proceedings of the 17th European Simulation Multiconference ESM-2003*, Nottingham, England, 2003, pp. 272–277.
- [29] R. Santana, E.P. de León, An evolutionary optimization approach for detecting structures on graphs, in: Dagli, Akay, Buczac, Ersoy, Fernandez (eds.), *Smart Engineering System Design: Neural Network, Fuzzy Logic, Rough Sets and Evolutionary Programming*, ASME press, 1998, pp. 371–376.
- [30] R. Santana, P. Larrañaga, J.A. Lozano, Side chain placement using estimation of distribution algorithms, *Artificial Intelligence in Medicine* 39 (1) (2007) 49–63.
- [31] R. Santana, H. Mühlenbein, Blocked stochastic sampling versus estimation of distribution algorithms, *Proceedings of the 2002 Congress on Evolutionary Computation CEC-2002*, vol. 2, IEEE press, 2002, pp. 1390–1395.
- [32] R. Santana, A. Ochoa, Dealing with constraints with estimation of distribution algorithms: the univariate case, in: A. Ochoa, M.R. Soto, R. Santana (eds.), *Proceedings of the Second Symposium on Artificial Intelligence (CIMA-99)*, Havana, Cuba, March 1999, pp. 378–384.
- [33] R. Santana, A. Ochoa, M.R. Soto, Factorized Distribution Algorithms for functions with unitation constraints, in: *Evolutionary Computation and Probabilistic Graphical Models*, *Proceedings of the Third Symposium on Adaptive Systems (ISAS-2001)*, Havana, Cuba, March 2001, pp. 158–165.
- [34] S. Shakya, J. McCall, D. Brown, Using a Markov network model in a univariate EDA: an empirical cost-benefit analysis, in: H.-G. Beyer, U.-M. O'Reilly (Eds.), *Proceedings of Genetic and Evolutionary Computation Conference GECCO-2005*, ACM, Washington, D.C., USA, 2005, pp. 727–734.
- [35] J.L. Shapiro, Drift and scaling in estimation of distribution algorithms, *Evolutionary Computation* 13 (1) (2005) 99–123.
- [36] P.A. Simionescu, D. Beale, G.V. Dozier, Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm, *Journal of Mechanical Design* 128 (1) (2007) 108–115.
- [37] J. Sun, Q. Zhang, E.P. Tsang, DE/EDA: a new evolutionary algorithm for global optimization, *Information Sciences* 169 (3–4) (2005) 249–262.
- [38] L. Tan, D. Taniar, Adaptive estimated maximum-entropy distribution model, *Information Sciences* 177 (15) (2007) 3110–3128.
- [39] S. Wright, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in: *Proceedings of the Sixth International Congress on Genetics*, 1932, pp. 356–366.

- [40] Q. Zhang, On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm, *IEEE Transactions on Evolutionary Computation* 8 (1) (2004) 80–93.
- [41] Q. Zhang, H. Mühlenbein, On the convergence of a class of estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation* 8 (2) (2004) 127–136.
- [42] L. Zinchenko, M. Radecker, F. Bisogno, Multi-objective univariate marginal distribution optimisation of mixed analogue-digital signal circuits, in: D. Thierens et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2007*, vol. II, ACM Press, London, UK, 2007, pp. 2242–2249.