Matching Inductive Search Bias and Problem Structure in Continuous Estimation–of–Distribution Algorithms

Peter A.N. Bosman *

Centre for Mathematics and Computer Science (CWI), P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Jörn Grahl

University of Mannheim, Department of Logistics, Mannheim, Schloss, 68131 Mannheim, Germany

Abstract

Research into the dynamics of Genetic Algorithms (GAs) has led to the field of Estimation–of–Distribution Algorithms (EDAs). For discrete search spaces, EDAs have been developed that have obtained very promising results on a wide variety of problems. In this paper we investigate the conditions under which the adaptation of this technique to continuous search spaces fails to perform optimization efficiently. We show that without careful interpretation and adaptation of lessons learned from discrete EDAs, continuous EDAs will fail to perform efficient optimization on even some of the simplest problems. We reconsider the most important lessons to be learned in the design of EDAs and subsequently show how we can use this knowledge to extend continuous EDAs that were obtained by straightforward adaptation from the discrete domain so as to obtain an improvement in performance. Experimental results are presented to illustrate this improvement and to additionally confirm experimentally that a proper adaptation of discrete EDAs to the continuous case indeed requires careful consideration.

Key words: Estimation–of–distribution algorithms; Numerical optimization; Normal distribution; Convergence dynamics; Adaptive variance

* Corresponding author. Tel.: +31-20-592-4323; fax: +31-20-592-4199.
Email addresses: Peter.Bosman@cwi.nl (Peter A.N. Bosman), joern.grahl@bwl.uni-mannheim.de (Jörn Grahl).

1 Introduction

Estimation–of–Distribution Algorithms (EDAs) constitute a relatively novel, yet already established, branch of Evolutionary Computation (EC). EDAs were initially introduced into the field of EC to overcome some of the shortcomings of earlier discrete Evolutionary Algorithms (EAs), specifically the simple Genetic Algorithm (GA). The main operator of variation in EDAs is the estimation of a probability distribution based on the selected solutions and the subsequent drawing of new solutions from this distribution. The distribution constitutes the explicit and adaptive inductive search bias of the EDA.

Along the line of binary and discrete EDAs, much progress has been made in recent years (Etxeberria and Larrañaga, 1999; Harik, 1999; Pelikan, Goldberg and Cantú-Paz, 1999; Mühlenbein and Mahnig, 1999; Harik and Goldberg, 2000; Pelikan and Goldberg, 2001; Pelikan and Goldberg, 2003). Some problems that were notoriously hard for GAs have been solved efficiently using EDAs. The variety of problems successfully tackled also contains some real–world problems (Blanco, Larrañaga, Inza and Sierra, 2001; Sierra, Lazkano, Inza, Merino, Larrañaga and Quiroga, 2001; Bengoetxea, Larrañaga, Bloch, Perchant and Boeres, 2002; Ducheyne, De Wulf and De Baets, 2002; Blanco, Inza and Larrañaga, 2003). The results have motivated researchers to extend the EDA principle to the continuous domain (Sebag and Ducoulombier, 1998; Gallagher, Fream and Downs, 1999; Bosman and Thierens, 2000; Larrañaga, Etxeberria, Lozano and Peña, 2000; Bosman and Thierens, 2001a; Cho and Zhang, 2001; Shin, Cho, and Zhang, 2001; Ocenasek and Schwarz, 2002; Ahn, Ramakrishna and Goldberg, 2004). However, the results so far are not as encouraging as those obtained in the binary domain in terms of search efficiency. In fact, current continuous EDAs fail on some standard test problems of numerical and parameter optimization where other continuous EAs or even classical gradient–based algorithms do not fail (Bosman and Thierens, 2001b).

Recently, studies have been carried out that indicate that the problems mentioned above may yet be coped with. Yuan and Gallagher (2005) showed in an initial investigation that by artificially keeping the variance at a value of at least 1, certain problems could be solved by a continuous EDA that it otherwise was not able to solve. Ocenasek, Kern, Hansen and Koumoutsakos (2004) used a self-adaptation approach adopted from evolution strategies to scale the variance after the distribution estimation. The underlying continuous probability distribution is quite involved however. Our approach in this paper is a more principled contribution to solving the problem of premature convergence and to explain what goes wrong in the adaptation of EDAs from the discrete domain to the continuous domain. In this article, it is assessed which requirements a probability distribution has to meet in order to function properly as

a search distribution in EDAs. We argue that there is a fundamental and systematic difference between the discrete and the continuous domain. Generally speaking, in order to build efficient optimizers using the EDA principle, the induced bias in the form of the estimated probability distribution has to fit to the structure of the problem at hand. The central topic of this paper is to assess the discrepancies between the concept of problem structure in the discrete and continuous domain and to assess to which extent the probabilistic search bias can fit the problem structure in the continuous domain. We indicate that indeed compared to the discrete case there are additional issues that need to be addressed in the design of the continuous EDA to decrease the probability of failure. We also present a simple remedy to meet with the additional issues we identify and show on the basis of experimental results that consequently the optimization performance of the continuous EDA indeed improves substantially.

The remainder of this paper is organized as follows. In Section 2 we outline the concept of EDAs and provide insight into the main lessons learned from discrete GAs and EDAs. In Section 3 we then show how continuous EDAs can be constructed by conforming to the main concept of EDAs and estimating a probability distribution from the selected solutions. In our case, we use maximum–likelihood estimates of the normal distribution. We also point out how the main lessons learned from the discrete domain can be transferred to the continuous domain. We show that a more careful interpretation is required of these lessons for the proper design of continuous EDAs. We illustrate analytically and by experiments how and why continuous EDAs can indeed fail. Afterwards, in Section 4, we propose a straightforward remedy with virtually no additional computational overhead. It is shown in Section 5 that by virtue of our remedy the continuous EDA no longer fails on certain benchmark problems. We conclude this article in Section 6.

2 Estimation–of–Distribution Algorithms (EDAs)

Estimation–of–Distribution Algorithms (EDAs), introduced by Mühlenbein and Paaß (1996), are stochastic search strategies that maintain a set of candidate solutions, called the population, throughout the search. A solution is also called an individual. Each individual has an associated fitness value that measures its quality. The goal of the EDA is to find the individual of highest quality. An individual consists of a phenotype and a genotype. The phenotype is the physical appearance (i.e. the actual solution to the problem at hand) whereas the genotype is the genetic encoding of the individual. The genotype-phenotype mapping is called the representation of the problem (Rothlauf, 2002). The fitness of an individual is computed using the phenotype, whereas new solutions are built on the basis of the genotype.

In an EDA, the first population of candidate solutions is usually generated at random. The fitness of the individuals is evaluated and the better individuals are selected for variation. Selection favors solutions of higher quality. Its function is to push the population into promising regions of the search space. New candidate solutions are then generated by estimating a probability distribution from the selected solutions and by randomly drawing new samples from this distribution. This specific approach to performing variation differentiates EDAs from other optimization techniques. The newly generated individuals replace parts of the old population or the entire old population as a whole. The new population is evaluated and the better individuals are kept. This process is then iterated until a predefined convergence criterion is met:

(1) Select a collection of solutions $\boldsymbol{\mathcal{S}}$ from $\boldsymbol{\mathcal{P}}$
(2) Estimate a probability distribution from $\boldsymbol{\mathcal{S}}$
(3) Draw a collection of new solutions $\boldsymbol{\mathcal{O}}$ from the estimated distribution
(4) Replace some of the solutions in $\boldsymbol{\mathcal{P}}$ by solutions from $\boldsymbol{\mathcal{O}}$

Specifically step 3 is different from GAs. In GAs, new solutions are typically created by crossing over parts of the genotype between a small number of parent solutions (typically 2). Different EDAs use different probability distributions and many EDAs have been proposed in recent years. For an overview we refer the interested reader to the literature (Larrañaga and Lozano, 2001; Pelikan, Goldberg and Lobo, 2002; Bosman and Thierens, 2004).

3 Adapting discrete EDAs to continuous EDAs

3.1 Matching inductive search bias and problem structure

In general, for an optimization algorithm to be competent in solving a certain optimization problem, the search bias of the optimization algorithm has to fit the structure of the problem. The search bias of EDAs, exemplified by the probability distribution used, is inductive as it is learned during optimization. Now, if it is possible to approximate the probability distribution over the solution space that assigns a uniform probability distribution over all solutions with a quality at least as good as that of the worst selected individual, a highly efficient EDA can be constructed (Rastegar and Meybodi, 2005). This EDA finetunes the probability distribution each generation to represent ever more precisely and selectively the best solutions in the search space. For EDAs, therefore, the following two prerequisites are of specific importance:

(1) Adequacy of the class of probability distribution
The probability distribution must be able to assign solutions that have

a certain minimal quality, i.e. solutions that have specific properties, a high probability density. In other words, the *capacity* of the probability distribution must be adequate.

(2) Competence of the estimation procedure
Even if the capacity of the class of probability distribution used is adequate, proper exploitation of the structure of the optimization problem is only guaranteed if the estimation procedure is actually capable of configuring the parameters of the probability distribution in such a way that the high probability densities are actually assigned to solutions of a certain minimal quality. In other words, the estimation procedure must be *competent*.

Finally, it should be noted that for efficiency, an additional prerequisite is that the estimation procedure is efficient (i.e. of low–order asymptotic algorithmic complexity) in addition to being competent.

3.2 Discrete EDAs

3.2.1 Inductive search bias

Probability distributions for discrete spaces assign probabilities to specific settings of variables. Hence, any probability distribution can be expressed, ensuring an adequate capacity. By factorizing the probability distribution (Lauritzen, 1996; Friedman and Goldszmidt, 1996), not all combinations of settings for all variables need to explicitly be enumerated, but probabilities can be assigned to specific combinations for subsets of variables. Since factorizations are only a more efficient way of representing the probability distribution, the capacity is not affected. Using frequency counts to estimate the probabilities from data results in maximum–likelihood estimations that reveal statistical dependencies.

3.2.2 Problem structure

In the discrete domain, problem structure refers to a decomposition of the optimization problem into subproblems of smaller sizes (Goldberg, 2002). In other words, there are configurations of bits at specific locations, so–called Building Blocks (BB), that contribute significantly to the solution quality when present in a solution. These building blocks are commonly said to form partial solutions to the problem. Moreover, the knowledge of which bit–configurations at what locations cause a significant contribution to the solution quality is commonly referred to as *linkage information* (Harik and Goldberg, 1996).

3.2.3 Matching

The necessity of the joint appearance of configurations of bits causes statistical dependence of random variables when estimating the probability distribution of the configurations of the bits from a set of solutions that were selected on the basis of their quality. Using factorized probability distributions, these statistical dependencies can be modeled. In other words, a discrete EDA can store which configurations of bits should have a large probability of appearing jointly in a good solution because the capacity prerequisite from Section 3.1 is met.

It has to be noted however, that in accordance with the degree of the interactions between the bits, simple or more involved factorizations need to be used. If there are no interactions between the bits, meaning that the building block size is one, univariately factorized probability distributions in which each variable is modeled to be statistically independent of each other variable, have proven to be efficient when used in an EDA (Pelikan and Mühlenbein, 1998). In general however the size of the building blocks is larger. As the interactions between the bits get more complex, the possibilities for expressing statistical dependency relations in the probability distributions should increase accordingly (Thierens, 1999; Bosman and Thierens, 1999). To this end, Bayesian factorizations have been found to be a suitable choice (Pelikan et al., 1999; Mühlenbein and Mahnig, 1999; Pelikan and Goldberg, 2003) as they meet the capacity prerequisite and in addition, a greedy estimation procedure is often found to meet the competence prerequisite from Section 3.1.

Summarizing, in the discrete space the inductive bias of factorized probability distributions based on frequency counts can match the decomposability of the problem structure. In addition, assuming that the decomposability is of bounded complexity, the greedy estimation procedure that is commonly employed in discrete EDAs is both competent and efficient. Consequently, discrete EDAs allow for efficient optimization.

3.3 Continuous EDAs

3.3.1 Inductive search bias

In the continuous domain, it is the contour–lines of the probability distributions that indicate which parts of the search space have a higher probability of being sampled.

3.3.2 Problem structure

Analogous to the inductive search bias, the problem structure in the continuous domain is exemplified by the contour–lines of the function to be optimized.

3.3.3 Matching

To match inductive search bias and problem structure, we thus need to match the contour lines. However, because the contour–lines of the optimization problem can be of virtually any shape, we require the property of universal approximation. However, such universal approximation is computationally intractable to compute. In practice, a continuous EDA will therefore have to rely on tractable probability distributions, such as ones that are based on the normal pdf.

Because in general we cannot assume that the contours of the fitness function can be modeled properly, a problem arises. The concept of statistical dependence no longer corresponds with dependence as imposed by the fitness landscape. For instance when using the normal pdf, after estimating the parameters it might be found that there is no statistical dependence between two variables. However, when observing the actual source for the data, which follows the density contours of the fitness function, the variables may be strongly dependent through non–linear interactions that simply cannot be modeled by the normal pdf. Hence, we can now conclude the following implications for the design of continuous EDAs regarding adequacy:

(1) Adequacy of the class of probability distribution (continuous domain)
   (a) *Adequate class of probability distribution*
   Linkage information in the continuous case only maps perfectly onto statistical dependency information as observed in the factorization of the probability distribution if the class of probability distribution that is used to perform density estimation with has the capacity to allow for a close modeling of the contours of the fitness landscape.
   (b) *Inadequate class of probability distribution*
   If there is a mismatch between the capacity of the class of probability distribution used for estimation and the contours of the fitness landscape, the modeling of statistical dependencies through factorizing the probability distribution in estimating the distribution from data is a less important and less reliable source of information for inducing the search bias in an attempt to exploit the structure of the problem.

From these revised prerequisites it follows that in the EDA based on the normal pdf it appears not to be a promising way to approach probabilistic modeling in the continuous domain with the same goal as in the discrete domain: to focus solely on getting the statistical dependencies right in the estimated

model and thereby assume a proper problem decomposition. Indeed, in initial EDAs that estimate a Bayesian factorization of the normal distribution using maximum–likelihood estimations, problems of inefficiency were already revealed after performing experiments on a variety of problems with differing problem structures (Bosman, 2003). It was observed that these EDAs are not capable of exploiting gradient information since density estimation makes no assumption on the source of the data from which to make the estimations. As a result, these EDAs were found to be extremely inefficient on problems with strong non–linear interactions between the variables, even in the presence of smooth gradients and unimodality. Similar results were found by other researchers, even using different probability distributions than the normal distribution, but still attempting to obtain a maximum–likelihood estimate, which is highly unlikely to capture the complete structure of the interesting part of the search space (Larrañaga et al., 2000; Cho and Zhang, 2001; Shin et al., 2001; Shin and Zhang, 2001; Cho and Zhang, 2002; Ocenasek and Schwarz, 2002; Paul and Iba, 2003a; Paul and Iba, 2003b; Ahn et al., 2004; Kern, Müller, Hansen, Büche, Ocenasek and Koumoutsakos, 2004; Cho and Zhang, 2004). As a consequence of the approach taken, premature convergence can occur even on very simple functions (Grahl, Minner and Rothlauf, 2005; Hansen, 2006). In the following section we move to mathematically quantify the apparent problems that we have now argued to come from a mismatch between the capacity of the class of probability distribution used and the contours of the fitness landscape.

3.4 A case study in continuous EDAs: the normal pdf

To gain further insight into the possible consequences of a mismatch between inductive search bias and problem structure in continuous EDAs, we propose to view the structure in a continuous search space as an arrangement of slopes and peaks. This aggregated view on the composition of a fitness landscape is also common in Evolution Strategies (ES, see for instance (Schwefel, 1995)). The probability distribution in an EDA has to fit to these fundamental elements. Since the normal pdf can only be properly matched to the contours of a single peak, it is important to note that slopes and peaks are related to each other. At the beginning of the search, the EDA will in general be approaching a local or global optimum on a slope or at least on a region of the search space that guides into the direction of a local or global optimum and thus has a slope–like shape. Whenever the search focuses around an optimum in its final phases, the current region will eventually be shaped like a peak. The use of the normal pdf will then fit the remaining local contours of the problem much better and efficient optimization should be possible according to the lessons mentioned above. An important question that remains is what happens when during the search the solutions are not (yet) concentrated around a peak.

3.4.1 The normal pdf and normal EDAs

Real–valued, continuous EDAs using Bayesian factorizations of the normal pdf were first researched by Bosman and Thierens (2000) and Larrañaga et al. (2000). Although these EDAs were among the very first to be studied for the continuous case, there are earlier works in the literature on continuous EDAs (Rudlof and Köppen, 1996; Servet, Trave-Massuyes and Stern, 1997; Sebag and Ducoulombier, 1998; Gallagher et al., 1999). The EDAs in these works use the normal pdf or a composition thereof, but allow no interactions to be taken into account between the problem variables, i.e. univariate probability densities are estimated. The common use of the normal pdf in EDAs is due to the computational tractability of the approach to estimating probability distributions in continuous spaces.

The normal pdf $P^{\mathcal{N}}_{(\boldsymbol{\mu_i},\boldsymbol{\Sigma^i})}$ for random variables $X_{\boldsymbol{i}}$ is parameterized by a vector $\boldsymbol{\mu_i}$ of means and a symmetric covariance matrix $\boldsymbol{\Sigma^i}$ and is defined by

$$P^{\mathcal{N}}_{(\boldsymbol{\mu_i},\boldsymbol{\Sigma^i})}(X_{\boldsymbol{i}})(\boldsymbol{x}) = \frac{(2\pi)^{-\frac{|\boldsymbol{i}|}{2}}}{(\det \boldsymbol{\Sigma^i})^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \left(\boldsymbol{\Sigma^i}\right)^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})} \tag{1}$$

The number of parameters to be estimated from data to fit the normal distribution to selected individuals equals $\frac{1}{2}|\boldsymbol{i}|^2 + \frac{3}{2}|\boldsymbol{i}|$. Different from the discrete case, the number of parameters to be estimated therefore does not grow exponentially with $|\boldsymbol{i}|$ but quadratically. As a result, estimating factorizations based on the normal pdf is relatively fast and efficient. For a given factorization, the parameters of the normal pdf have to estimated from the selected individuals. A maximum–likelihood estimation for the normal pdf is obtained from a vector $\boldsymbol{\mathcal{S}}$ of samples if the parameters are estimated by the sample average and the sample covariance matrix (Anderson, 1958; Tatsuoka, 1971):

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \frac{1}{|\boldsymbol{\mathcal{S}}|}\sum_{j=0}^{|\boldsymbol{\mathcal{S}}|-1} (\boldsymbol{\mathcal{S}}_j)_{\boldsymbol{i}}, \qquad \hat{\boldsymbol{\Sigma}}^{\boldsymbol{i}} = \frac{1}{|\boldsymbol{\mathcal{S}}|}\sum_{j=0}^{|\boldsymbol{\mathcal{S}}|-1} ((\boldsymbol{\mathcal{S}}_j)_{\boldsymbol{i}} - \hat{\boldsymbol{\mu}}_{\boldsymbol{i}})((\boldsymbol{\mathcal{S}}_j)_{\boldsymbol{i}} - \hat{\boldsymbol{\mu}}_{\boldsymbol{i}})^T \tag{2}$$

Note that to estimate Bayesian factorizations using the normal pdf, a way to estimate the parameters for the univariate conditional normal pdf is required. Fortunately, these parameters can be efficiently computed using the maximum–likelihood estimations in Equation 2. For more details see (Bosman and Thierens, 2000) or (Bosman, 2003).

Although more involved probability distributions can be defined and used in EDAs using for instance mixtures of normal pdfs (Bosman and Thierens, 2001a; Ahn et al., 2004), we specifically focus on the use of the single normal distribution in this article as it is more intuitive to analyze. Moreover, the use of the less–involved normal pdf will not prevent us from obtaining a better

understanding of the exploitation of problem structure using continuous EDAs and to reflect in a general manner on the lessons learned from the discrete domain. In the remainder of this section we turn our attention to exactly these specific topics.

3.4.2 The inductive search bias on a slope

In this section, we analyze analytically how a slope is traversed by a simple one–dimensional EDA that employs maximum–likelihood normal density estimation and sampling. Without loss of generality, we assume that the fitness should be maximized.

Our analysis is based on a regular EDA run as defined in Section 2. The population size is assumed to be n. The fitness function f is monotonous, modeling a slope. In the selection step, the best $\lfloor \tau n \rfloor$ solutions are selected, $\tau \in [\frac{1}{n}, 1]$. A univariate normal density is estimated with maximum likelihood from the selected individuals. From this density, n new individuals are generated. The new individuals replace the old population completely. We are interested in how the population statistics μ^t (mean) and $(\sigma^t)^2$ (variance) change as generations pass.

3.4.2.1 Monotonous fitness functions Let $\boldsymbol{\mathcal{S}}$ be a set of solutions. Let x_j and $x_k \in \boldsymbol{\mathcal{S}}$ be two distinct solutions and let $g : \mathbb{R} \to \mathbb{R}$ be a fitness function. Then:

$$\begin{aligned} &g \text{ is increasing if } x_j \leq x_k \text{ implies that } g(x_j) \leq g(x_k)\ \forall x_j, x_k \in S \\ &g \text{ is decreasing if } x_j \leq x_k \text{ implies that } g(x_j) \geq g_i(x_k)\ \forall x_j, x_k \in S \end{aligned} \tag{3}$$

The fitness landscape is said to be monotonous if the fitness function f is either increasing or decreasing.

Assume now that a population $\boldsymbol{\mathcal{P}}$ of search points is given. We use m different increasing functions $f_0, f_1 \ldots f_{m-1}$ to evaluate the solutions in population $\boldsymbol{\mathcal{P}}$. After each evaluation process, we use truncation selection to select the best $\lfloor \tau n \rfloor$ individuals. We denote the m sets of selected individuals by $\boldsymbol{\mathcal{S}}_0, \boldsymbol{\mathcal{S}}_1, \ldots, \boldsymbol{\mathcal{S}}_{m-1}$. It is a simple, yet interesting fact that all sets $\boldsymbol{\mathcal{S}}_i$ have to be identical. The EDA under study uses density estimation and sampling to generate new candidate solutions. In the density estimation process, the fitness of the individuals is not considered. Density estimation solely relies on the location of the points, i.e. the value of x. As the parameters μ^t and $(\sigma^t)^2$ are estimated from x, they are identical for all f_i.

This fact simplifies our further analysis. We can now state that the effects of iterated density estimation and sampling from the normal pdf will be the same for all increasing fitness functions (and for all decreasing functions). Thus, we can base our analysis on the simplest monotonous function, which is the linear one, because we know that our results are valid for all monotonous functions.

3.4.2.2 Truncation selection and monotonous fitness functions We analyze the truncation selection step in the presence of a monotonous fitness function. As stated above, the EDA generates new candidate solutions by sampling x from a normal distribution with mean μ^t and variance $(\sigma^t)^2$. Assume now that the fitness y is obtained from a linear function $y = a \cdot x + b$. In this case, we can even further simplify our analysis. This is due to using truncation selection. Truncation selection selects the best $\lfloor \tau n \rfloor$ individuals, regarding only the fitness ranks of the individuals. The fitness ranks are however independent of a and b. We can therefore choose $y = x$ as the simplest linear function.

Now, we are interested in the individual $x_{\min}$, that is, the individual with minimal fitness out of all selected individuals. All individuals $x > x_{\min}$ are selected. We call $x_{\min}$ the population truncation point. The population truncation point $x_{\min}$ can be obtained as follows:

$$x_{\min} = \Phi^{-1}_{\mu^t,\sigma^t}(1-\tau), \tag{4}$$

where $\Phi^{-1}_{\mu^t,\sigma^t}$ is the inverse cumulative distribution function of a normal distribution with mean μ^t and standard deviation σ^t. Thus, for monotonous fitness functions and truncation selection, the selected individuals can be directly obtained from the population statistics.

3.4.2.3 Change of μ^t to μ^{t+1} We model selection by truncation of the normally distributed population density. Assume that a population is distributed with mean μ_i^t and standard deviation σ^t. The fitnesses of the individuals are calculated and the best $\lfloor \tau n \rfloor$ solutions are selected. This corresponds to a truncation of the normal distribution from the left at $x_{\min}$. Let $\phi(x)$ be the standard normal density at value x. From econometric literature on the truncated normal distribution (see (Greene, 2003), appendix) we have that the mean of a left–truncated normal distribution truncated in $x_{\min}$ is:

$$E(X|X > x_{\min}) = \mu + \sigma \cdot \frac{\phi(\frac{x_{\min}-\mu}{\sigma})}{\Phi(\frac{x_{\min}-\mu}{\sigma})} \tag{5}$$

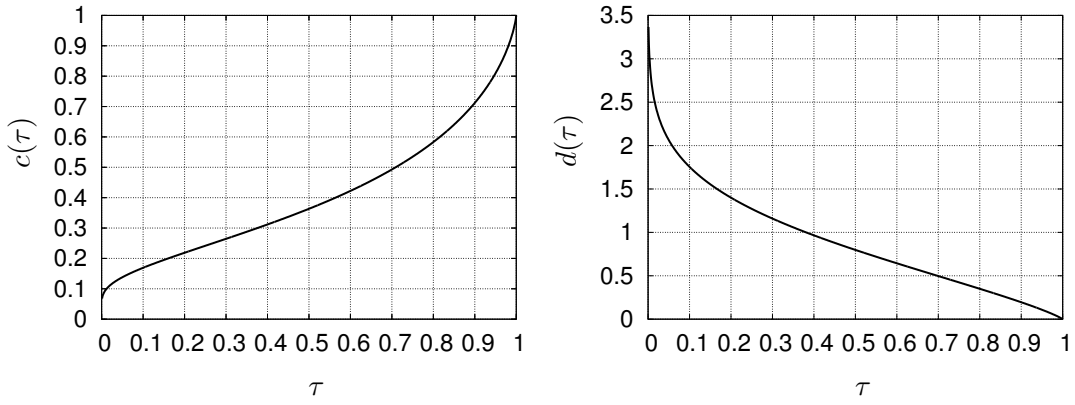

Fig. 1. Illustration of $c(\tau)$ and $d(\tau)$

Now, inserting the result from Equation 4 and rearranging leads to:

$$\mu^{t+1} = E(X|X > x_{\min}) = \mu^t + \sigma^t \cdot \frac{\phi\left(\Phi^{-1}(\tau)\right)}{\tau} = \mu^t + \sigma^t \cdot d(\tau) \tag{6}$$

$$\text{where } d(\tau) = \frac{\phi\left(\Phi^{-1}(\tau)\right)}{\tau}$$

The mean of the population after applying truncation selection can now be easily computed. The factor $d(\tau)$ is illustrated in Figure 1. It can be seen that for $\tau \to 1$ the factor $d(\tau)$ converges to 0. In this case the mean of the population remains unchanged, i.e. $\mu^t = \mu^{t+1}$.

3.4.2.4 Change of σ^t to σ^{t+1} Again, we model truncation selection by truncation of the normally distributed population density. The variance of a normal distribution that is left-truncated in $x_{\min}$ is given by:

$$Var(X|X > x_{\min}) = E(X^2|X > x_{\min}) - E(X|X > x_{\min})^2 \tag{7}$$

$$= \sigma^2 \cdot \left\{ 1 + \frac{\frac{x_{\min}-\mu}{\sigma} \cdot \phi\left(\frac{x_{\min}-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{x_{\min}-\mu}{\sigma}\right)} - \left[\frac{\phi\left(\frac{x_{\min}-\mu}{\sigma}\right)}{1 - \Phi\left(\frac{x_{\min}-\mu}{\sigma}\right)} \right]^2 \right\}$$

We use this equation in the context of our model by assigning appropriate indices, inserting $x_{\min}$, simplifying, and rearranging. This leads us to:

$$(\sigma^{t+1})^2 = (\sigma^t)^2 \cdot c(\tau) \tag{8}$$

$$\text{where } c(\tau) = \left\{ 1 + \frac{\Phi^{-1}(1-\tau)\phi\left(\Phi^{-1}(\tau)\right)}{\tau} - \left[\frac{\phi\left(\Phi^{-1}(\tau)\right)}{\tau} \right]^2 \right\}$$

Now, we can compute the population variance in generation $t+1$, given the population variance in generation t. The factor $c(\tau)$ is plotted in Figure 1. It

can be seen that if $\tau \to 1$, the factor $c(\tau)$ converges to 1. In this case the variance of the population remains unchanged, i.e. $(\sigma^t)^2 = (\sigma^{t+1})^2$.

3.4.2.5 Population statistics in generation t Ultimately, we are interested in how the population mean and variance depend on t. To obtain the corresponding population statistics, Equations 6 and 8 from Sections 3.4.2.3 and 3.4.2.4 must be repeatedly used. By doing so one obtains the following result for the mean in generation t:

$$\mu^t = \mu^0 + \sigma^0 \cdot c(\tau) \cdot \sum_{i=1}^{t} \sqrt{d(\tau)^{i-1}} \tag{9}$$

Similarly, one obtains the following result for the variance in generation t:

$$(\sigma^t)^2 = (\sigma^0)^2 \cdot c(\tau)^t \tag{10}$$

3.4.2.6 Convergence of population statistics for $t \to \infty$ In this section, we analyze the convergence of the EDA. To do so, we analyze how the population statistics develop over time as $t \to \infty$. First, we consider the mean. Therefore, we make use of (9). Note that the sum is the only part of the expression that depends on t. This leads us to:

$$\begin{aligned} \lim_{t\to\infty} \mu^t &= \mu^0 + \sigma^0 \cdot c(\tau) \cdot \underbrace{\lim_{t\to\infty} \sum_{k=1}^{t} \left[\sqrt{d(\tau)^{(k-1)}}\right]}_{\text{infinite geometric series}} \\ &= \mu^0 + \sigma^0 \cdot c(\tau) \cdot \frac{1}{1 - \sqrt{d(\tau)}} \end{aligned} \tag{11}$$

This expression allows us to compute the maximum distance that the population mean can move across the search space for a given selection percentile τ and under the assumption of a monotonous fitness function. Also, this expression indicates that this maximum distance is bounded.

Now, we consider the variance. We make use of (10) and let t tend to infinity. Note that $0 < c(\tau) < 1$. This leads to

$$\lim_{t\to\infty} (\sigma^t)^2 = \lim_{t\to\infty} \left[(\sigma^0)^2 \cdot c(\tau)^t\right] = 0 \tag{12}$$

Thus, the variance converges to 0.

3.4.2.7 Interpretation of results From the previous paragraph we now know that the continuous EDA based on maximum–likelihood estimations of the normal distribution converges even while on a slope since the population variance converges to 0. The reason for this is that the maximum distance that the mean of the population can move across the search space is bounded. The final position of the mean solely depends on

- the mean of the first population,
- the variance of the first population,
- and the selection percentile τ.

It is thus well possible that the EDA is unable to find the global optimum. If at some point during optimization solutions are only available on a slope, but the optimum is (well) outside of the range of the current set of solutions, the algorithm will converge prematurely on its way toward the optimum. Note that this failure will not vanish when switching to higher dimensional search spaces. Also note that the loss of solutions surrounding the optimum can easily occur during a run of the EDA as the normal pdf that is used is likely not to match the problem structure. Hence, efficient maintenance of solutions surrounding an optimum is not guaranteed, following the prerequisites from Section 3.1, specifically prerequisite 1. Since the maximum–likelihood estimation procedure lacks the possibility of generalization to fit the search space outside of the area defined by the selected solutions, prerequisite 2 is not met with either. From the previous section we now know that the price to pay is likely to be that of premature convergence.

An experimental illustration of optimization failure is presented in Figure 2. The Figure shows the result of using a one–dimensional maximum–likelihood normal EDA to minimize the sphere function. The progression of density estimations is shown in subsequent generations. The solutions are initially in the range $[-10; -5]$. Indeed, even though the function to optimize has a smooth gradient and is unimodal, the EDA is not able to find the optimum because the variance goes to zero too rapidly. The problem caused by lack of generalization is immediately apparent from this Figure.

The resulting situation is comparable to the loss of building blocks during a GA run in the discrete domain. Discrete GA theory tells us that in that case the population size should be increased. However, because of the limiting shape of the density function to be estimated we know that increasing the population size will not help to obtain a better approximation of the true density. Hence, the population size will have to increase dramatically to improve the initial quality of solutions. Under the use of elitism, these solutions will then be maintained throughout the run, increasing the eventual possibility of ending up with a distribution of solutions surrounding a peak. However, such increase will be exponential in the number of variables due to the curse of di-

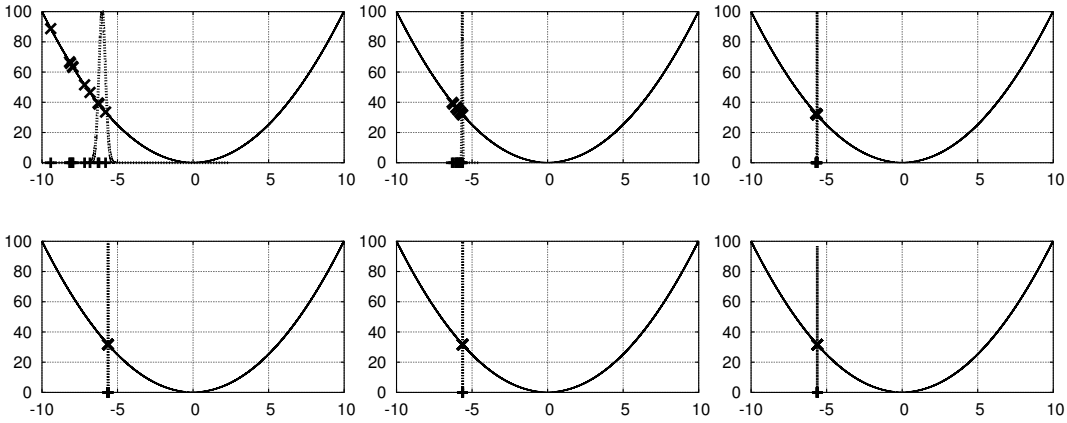

Fig. 2. Population and estimated probability distribution (rescaled to fitness range for visualization) in the maximum–likelihood normal EDA in generations 0, 1, 2, 3, 4 and 5 (top–left to bottom–right).

mensionality. Moreover, if the optimum is simply not contained in the initial range in which samples are available, increasing the population size will not improve the probability of finding the optimum at all.

Concluding, using maximum–likelihood estimations of normal pdfs, the search bias cannot be fit to match the structural properties of slope–like regions in the fitness landscape due to lack of generalization.

3.4.3 The inductive search bias on a peak

Assuming that solutions are distributed nicely surrounding a peak in the landscape, it is evident that the search bias induced by an EDA based on the normal pdf will fit the problem structure well. The reason is that the unimodality of the normal pdf will place the center of mass of the estimated distribution near the true center of the landscape, increasing the probability of generating solutions near the optimum.

4 Enhancing maximum–likelihood normal EDAs using adaptive variance scaling and correlation triggering

In the previous section we have analyzed how the induced bias of the normal pdf fits to two elementary structures of a continuous search space: slopes and peaks. We found, that the induced search bias cannot be made to fit the structure of a slope well enough to guarantee successful search, whereas it imposes no problem on peaks. As pointed out, both structures will, however,

in general appear during an EDA run. Since we are not interested in making the class of probability distribution more involved in this article, the most important question that now arises is how the estimation procedure in the normal EDA should be changed to prevent the identified problems as best possible.

In this section we first introduce a simple technique that modifies the estimation procedure of the normal pdf in a way that makes it more effective when traversing a slope. Subsequently we propose a triggering method that allows to decide during optimization whether the use of this efficiency enhancement is currently appropriate or not.

4.1 Adaptive variance scaling

The smaller the variance in the estimated probability distribution, the smaller the area of exploration for the EDA. The variance in the normal pdf is explicitly stored in the covariance matrix $\mathbf{\Sigma}$. Hence, a straightforward manner to allow the EDA to increase the area of exploration is to enlarge the variance beyond its maximum–likelihood estimate.

The rationale that we propose for the actual scaling of the variance of the estimated normal pdf is the following. An adaptive–variance–scaling coefficient c^{AVS} is maintained. Upon drawing new solutions from the probability distribution, the variance is scaled by c^{AVS}, i.e. the covariance matrix used for the normal pdf is $c^{\text{AVS}}\mathbf{\Sigma}$ instead of just $\mathbf{\Sigma}$. If the best fitness value improves in one generation, the current size of the variance allows for progress. Hence, a further enlargement of the variance may allow for further improvement in the next generation. Note that this rationale is in accordance with lesson 1b from Section 3.3: since the estimation procedure of the normal pdf can in general not be expected to be the most reliable source of fitting the problem structure, we incorporate additional sources of information. The rationale is also in accordance with lesson 2 from Section 3: if the search has reached a point where all solutions are on the same slope, the maximum–likelihood estimation procedure will not generalize to parts outside of the datarange where the fitness values may also be interesting. Enlarging the variance helps in tackling this problem. In this case we observe the actual result in improvement of using a certain estimation. To compensate for the variance–diminishing effect of selection, the size of c^{AVS} is scaled by $\eta^{\text{Inc}} > 1$. If on the other hand the best fitness does not improve, the range of exploration may be too large to be effective and the adaptive–variance–scaling coefficient should be (slowly) decreased by a factor $\eta^{\text{Dec}} \in (0; 1)$. To allow for the effect of adaptive variance scaling to extend over multiple generations, we propose to not let c^{AVS} decrease too rapidly, e.g. set $\eta^{\text{Dec}} = 0.9$. For symmetry, we set $\eta^{\text{Inc}} = 1/\eta^{\text{Dec}}$.

We bound the magnitude of c^{AVS} from above by a predefined value $c^{\text{AVS–MAX}}$ to ensure that improvements in the best fitness in many subsequent generations do not lead to excessive variance values. We also bound the magnitude of c^{AVS} from below by a predefined value $c^{\text{AVS–MIN}}$. Whereas from the results in Section 3 it can be argued that the only problem with the EDA approach based on the maximum–likelihood normal distribution is that the variance can become too small, allowing the variance to be scaled to even smaller values can speed up convergence when the EDA seems not able to find better solutions for a sustained number of subsequent generations (often due to multiple local minima). For symmetry we set $c^{\text{AVS–MIN}} = 1/c^{\text{AVS–MAX}}$. Moreover, whenever it happens that $c^{\text{AVS}} < c^{\text{AVS–MIN}}$, decreasing the size of the variance doesn't appear to help and c^{AVS} is reset to $c^{\text{AVS–MAX}}$ to stimulate exploration. An experimental illustration of the normal EDA extended with the adaptive–variance–scaling technique is presented in Figure 3. Indeed the EDA is now capable of finding the optimum even though it is outside of the initial sampling range.

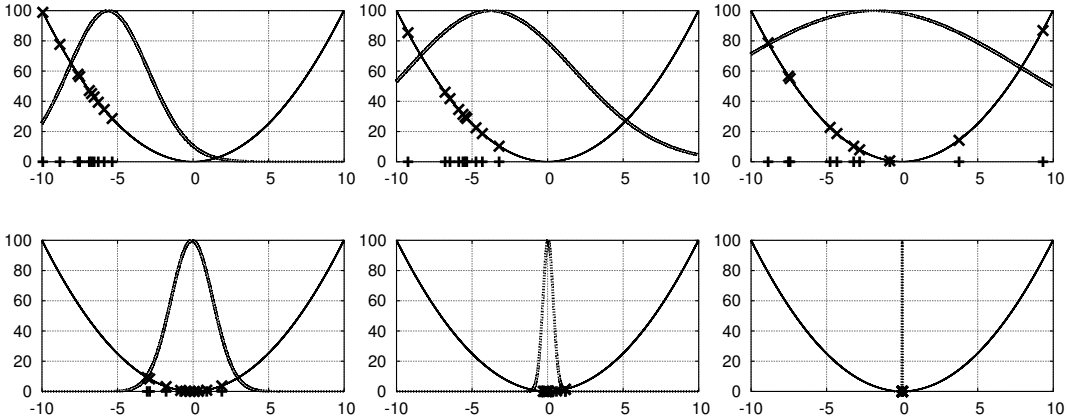

Fig. 3. Population and estimated probability distribution (rescaled to fitness range for visualization) in the adaptive–variance–scaling maximum–likelihood normal EDA in generations 0, 1, 2, 4, 8 and 16 (top–left to bottom–right).

4.2 Correlation–triggered adaptive–variance-scaling

In the scheme defined in Section 4.1 the adaptive–variance–scaling coefficient c^{AVS} increases if a better fitness value is found, i.e. if the EDA is successful in a certain generation. A success does however not always mean that the variance needs to be enlarged. This is especially the case when the center of the normal pdf is close to the optimum. Once this is the case, the induced bias of the normal pdf suffices to guide the search to the optimum. Making the variance larger in such a case will only slow the EDA down as it leads the bias of the algorithm to also explore a larger area around the optimum. Because this essentially makes the EDA less efficient, adaptive variance scaling

is to be prevented in such a case. Note that this approach to distinguishing between the two situations during the EDA run is actually a test that indicates whether the currently induced search bias suits the structure of the current search area. If it does, the maximum–likelihood probabilistic modeling of the normal pdf can be used (following the combination of prerequisites 1a and 2). Otherwise, additional means of inducing the search bias may be extremely helpful (following the combination of prerequisites 1b and 2).

To obtain a test of the reliability of using structure identification in continuous EDAs by means of maximum–likelihood estimations, the relationship between normal density and fitness of the selected solutions can be exploited. If the selected solutions are centered around a (local) optimum, the density will be strongly correlated with fitness (positively in case of maximization and negatively in case of minimization). The reason is that for the normal distribution the density of a point decreases when it is moved away from the mean. Intuitively, such correlation is desirable since better fitness values get a higher probability of being (re)produced by the EDA. If on the other hand the selected solutions are found to be on a slope, on the one side of the mean the fitness values will be better whereas on the other side of the mean the fitness values will be worse. Hence, a decrease in density is associated with both an increase and a decrease in fitness, effectively decorrelating density and fitness.

We propose to base the test for triggering the use of adaptive variance scaling on the ranked correlation coefficient between density and fitness. We use ranked correlation because the most important aspect is that a larger density should be associated with a better fitness value whereas the exact form of the fitness landscape is less important. The results of using this correlation trigger for a slope and for a peak are illustrated in Figure 4.

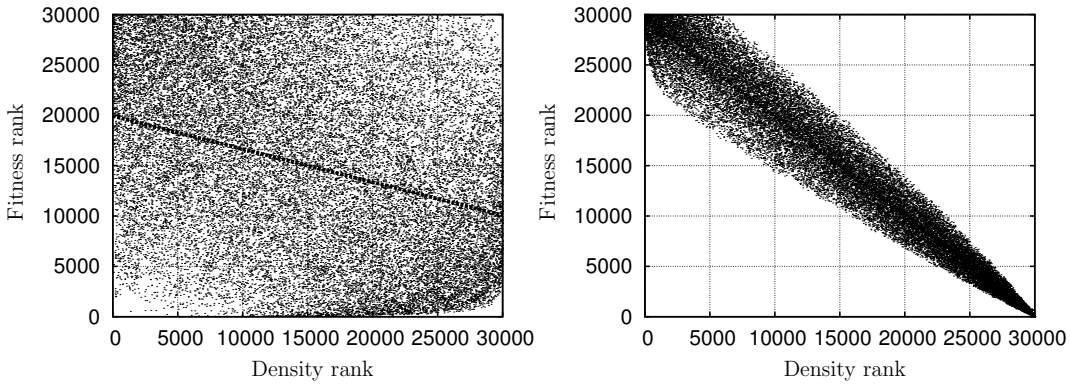

Fig. 4. Scatterplots and corresponding regression lines for fitness of the selected solutions versus their density under the estimated normal distribution in the first generation when minimizing the sphere function for $l = 5$. *(Left)* initial range $= [-10, -5]$ ($r = -0.3289859$), *(Right)* initial range $= [-3, 2]$ ($r = -0.9725636$).

We propose to have a threshold value θ^{corr} such that if the value of the correlation coefficient r between the density and the fitness of the selected solutions

is at most the value of the threshold, i.e. $\theta^{\text{corr}} \leq r$, then the conventional maximum–likelihood estimate is used in the EDA. Otherwise, the estimate based on adaptive variance scaling is used. Note that in the case of maximization we should test for $\theta^{\text{corr}} \geq r$ instead. An experimental illustration of using adaptive variance scaling only when the correlation test was not passed is presented in Figure 5. Indeed, adaptive variance scaling is now not always used, preventing the variance from becoming unnecessarily large and speeding up convergence.

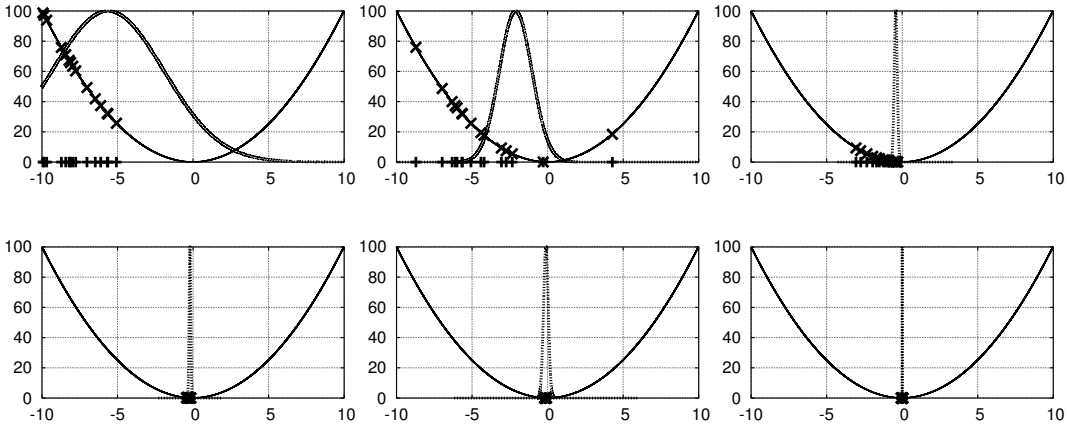

Fig. 5. Population and estimated probability distribution (rescaled to fitness range for visualization) in the correlation–triggered adaptive–variance–scaling maximum–likelihood normal EDA in generations 0, 1, 2, 4, 6 and 8 (top–left to bottom–right).

5 Experiments

In this section we present the results of experiments performed with the various EDAs based on the normal pdf as described earlier. For clarity, the base normal EDA that we use, employs the greedy building of a Bayesian factorization of the normal pdf with maximum–likelihood estimates as initially introduced under the acronym IDEA (Bosman and Thierens, 2000). The experiments serve to indicate whether our closer analysis of the dynamics of continuous EDAs and their subsequent redesign indeed leads to more efficient continuous EDAs and hence supports our adjusted formulation of lessons and prerequisites for the design of EDAs. For comparison with other existing optimization techniques, we used the state-of-the art in evolution-strategies research, i.e. the CMA-ES (Hansen, Muller and Koumoutsakos, 2003; Kern et al., 2004). We further used six well–known numerical optimization problems. The dimensionality of these problems was varied to get a total of twelve problem instances to test the algorithms on.

5.1 Optimization problems

Our test suite consists of six problems that are defined for a general dimensionality of l. These problems represent a variety of difficulties in numerical optimization. The definitions of the problems are given in Table 1.

Name	Definition	Initial range	Optimal value
Sphere	**Minimize** $\sum_{i=0}^{l-1} x_i^2$	$x_i \in [-5, 5]$ $(0 \leq i < l)$	0
Parabolic Ridge	**Minimize** $-x_0 + 100 \sum_{i=1}^{l-1} x_i^2$	$x_i \in [-5, 5]$ $(0 \leq i < l)$	$-\infty$
Griewank	**Minimize** $\frac{1}{4000} \sum_{i=0}^{l-1} (x_i - 100)^2 - \prod_{i=0}^{l-1} \cos\left(\frac{x_i - 100}{\sqrt{i+1}}\right) + 1$	$x_i \in [-600, 600]$ $(0 \leq i < l)$	0
Michalewicz	**Minimize** $-\sum_{i=0}^{l-1} \sin(x_i) \sin^{20}\left(\frac{(i+1)x_i^2}{\pi}\right)$	$x_i \in [0, \pi]$ $(0 \leq i < l)$	-4.687658 (l = 5) -24.63319 (l = 25)
Rosenbrock	**Minimize** $\sum_{i=0}^{l-2} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$x_i \in [-5.12, 5.12]$ $(0 \leq i < l)$	0
Summation Cancellation	**Maximize** $100/(10^{-5} + \sum_{i=0}^{l-1} \lvert\gamma_i\rvert)$ **where** $\gamma_0 = x_0, \gamma_i = x_i + \gamma_{i-1}$	$x_i \in [-3, 3]$ $(0 \leq i < l)$	$1 \cdot 10^7$

Table 1
Numerical optimization test problems (optimal values given with 7–digit precision).

The sphere function is probably the most standard unimodal benchmark for numerical optimization. The parabolic ridge function is a parabola, i.e. just like the sphere function, in all dimensions except the first one. In the first dimension it is a simple linear function. To find the optimum for this function, the value for the first variable therefore needs to be moved extremely far outside its initial range. Griewanks function is a function with many local optima. Basically it is a parabola superimposed with a sine function to obtain many local optima. Michalewicz's function is also a function with many local optima, albeit to a lesser degree than Griewanks function. An important difference is that Michalewicz's function has many long channels throughout which the minimum value is the same. Rosenbrocks function is highly non–linear. It has a very narrow and curved valley along which the quality of the solutions is much better than in its close neighborhood. This function is a real challenge for any numerical optimizer as it is very hard to capture the entire structure of the function and success is therefore only guaranteed if the gradient along the bottom of the valley is followed. The summation cancellation function has strong multivariate linear interactions between all problem variables.

5.2 Experiment setup

We varied the dimensionality for each problem to get an indication of the applicability of each algorithm as the number of problem variables increases. To

Dim.	Sphere	Par. Rid.	Griewank	Michalewicz	Rosenbrock	Sum. can.
$l = 5$	$1 \cdot 10^{-20}$	$-1 \cdot 10^{10}$	$1 \cdot 10^{-10}$	-4.687658	$1 \cdot 10^{-10}$	$1 \cdot 10^{7}$
$l = 25$	$1 \cdot 10^{-20}$	$-1 \cdot 10^{10}$	$1 \cdot 10^{-10}$	-24.63319	$1 \cdot 10^{-10}$	$1 \cdot 10^{7}$

Table 2
Values–to–reach for all problems and dimensionalities.

be precise, we used $l \in \{5, 25\}$. We ran tests for the normal EDA, the normal EDA with adaptive variance scaling and the normal EDA in which adaptive variance scaling is triggered by the ranked–correlation test. The baseline normal EDA uses greedy estimation of a Bayesian factorization of the normal pdf (Bosman and Thierens, 2001a). Furthermore, elitism is used in the sense that all selected solutions are kept every generation. The $n - \lfloor \tau n \rfloor$ non–selected solutions are replaced by drawing new solutions from the estimated probability distribution.

For $l = 5$, we enforced a maximum of $1 \cdot 10^6$ evaluations. For $l = 25$ we enforced a maximum of $5 \cdot 10^6$ evaluations. We ran tests for various population sizes ($\{25, 50, 100, 200, 400, 800, 1600, 3200, 6400, 12800\}$) to find the best results averaged over 100 independent runs. By "best result" we refer to the smallest population size for which the value–to–reach (see Table 2) was reached in all 100 runs. If this success was not obtained for any population size, the population size that resulted in the highest probability of success was used. If the probability of success for found to be 0, the average best fitness value was considered. For all EDAs we used the rule of thumb by Mühlenbein and Mahnig (1999) for FDA and set the selection threshold τ to 0.3. We used $\eta^{\text{DEC}} = 0.9$, i.e. a small multiplication factor to allow for smooth adaptation of the variance multiplication factor. The magnitude of c^{AVS} was bounded from above by $c^{\text{AVS-MAX}} = 10.0$.

5.3 Results

5.3.1 Parameter selection for the correlation trigger

In order to select a reasonable value for θ^{corr}, we tested when the ranked correlation coefficient between fitness and density actually triggers scaling of the variance on the sphere function. We varied θ^{corr} from -1.0 to 1.0 in steps of 0.01. For each value of θ^{corr}, 100 independent runs were performed on the sphere function in dimensionalities $l \in \{2, 4, 8, 10, 20, 40, 80\}$. Initial populations were drawn symmetrically around the optimal solution of 0 for all dimensions in a range of $[-7.5, 7.5]$. The population size that was used for a certain dimensionality l was equal to the minimally required population size for the EDA without variance adaptation to solve the problem with dimensionality l to optimality. In other words, in that case variance scaling is not required because

the sphere function is a single peak and thus, the induced bias of the normal pdf itself suffices to locate the optimum if the population size is large enough.

Figure 6 illustrates the percentage of generations in which variance scaling was nonetheless triggered (averaged over 100 runs). As a rule of thumb, we propose to set $\theta^{\text{corr}} = -0.55$. For this value, the number of unnecessary correlation triggers is rather constant and at most 25%. If a smaller value (i.e. closer to -1.0) is chosen, it can be seen from Figure 6 that the number of unnecessary correlation triggers will grow with increasing dimensionality. Although the value of -0.55 is rather robust, i.e. values between -0.6 and -0.4 lead to good results, the value for the correlation trigger should not become much larger. If a larger value (i.e. closer to 1.0) is chosen, the scaling of variances was observed from initial experimentation not to be triggered when it is required on slopes.

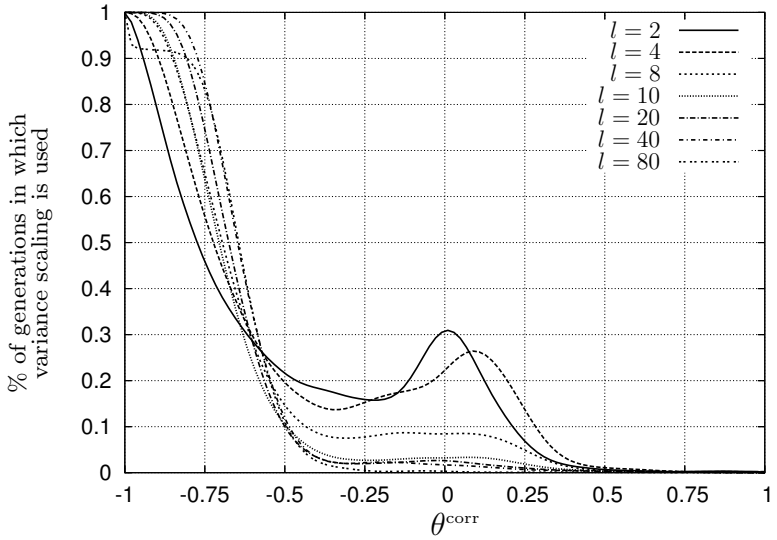

Fig. 6. Percentage of generations in which variance scaling is triggered unnecessarily on the sphere function in various dimensionalities as a function of the value used for the correlation trigger threshold θ^{corr}.

5.3.2 Overall results on the test–suite

In Figure 7 the average convergence behavior is shown for the normal EDA, for the normal EDA with adaptive variance scaling, for the normal EDA with adaptive variance scaling triggered by the ranked correlation coefficient and for the CMA–ES. The CMA–ES represents the state-of-the art in evolution–strategies research (Hansen, 2006). The population size that was selected according to the guidelines described in Section 5.2. In Table 3 the selected population sizes and associated probabilities of success, denoted Pr.(S), are tabulated. Since the optimal value for Michalewicz's function is negative and we wanted to present the results on a logarithmic scale we plotted the distance to the optimal value instead. Moreover, since the optimal value for the

parabolic ridge function is unbounded, we presented the results for this particular test problem on a linear scale. With the exception of the sphere problem and the parabolic ridge function, all graphs also have a logscale on the horizontal axis. Note that when a convergence graph indicates the value to reach was not obtained, this doesn't mean that this was the case in all 100 runs.

One of the things that immediately stands out from the results is the apparent inability of the normal EDA to optimize the parabolic ridge function and Rosenbrocks function, although both functions have nice smooth gradient properties. The probability of success within the bounds used in our experimentations is truly 0. Although the gradient along the direction toward the optimum is straightforward, i.e. it is a simple linear slope for the parabolic ridge function, the variance in the EDA without variance scaling shrinks too fast and the slope cannot be traveled. In general, the normal EDA isn't very successful. However, when it is successful (i.e. on the sphere function), it tends to be quite efficient. However, the overall drawback of using only the normal distribution with maximum–likelihood estimations is apparent. Even using very large population sizes, various problems cannot be solved at all.

EA	**Dim.**	**Sphere**		**Par. Rid.**		**Griewank**		**Michalewicz**		**Rosenbrock**		**Sum. can.**	
		n	Pr.(S)	n	Pr.(S)	n	Pr.(S)	n	Pr.(S)	n	Pr.(S)	n	Pr.(S)
Normal	$l = 5$	100	1.0	12800	0.0	800	0.0	100	0.2	12800	0.0	400	1.0
	$l = 25$	400	1.0	12800	0.0	400	1.0	800	0.0	12800	0.0	12800	0.03
Normal+	$l = 5$	25	1.0	25	1.0	800	1.0	50	0.21	50	1.0	50	1.0
AVS	$l = 25$	50	1.0	50	1.0	100	1.0	6400	0.01	100	1.0	800	1.0
Normal+	$l = 5$	50	1.0	50	1.0	800	1.0	200	0.3	50	1.0	50	1.0
AVS+CT	$l = 25$	50	1.0	50	1.0	100	1.0	12800	0.0	200	1.0	1600	1.0
CMA-ES	$l = 5$	25	1.0	25	1.0	200	1.0	25	0.05	25	1.0	25	1.0
	$l = 25$	25	1.0	25	1.0	50	1.0	200	0.0	25	1.0	25	1.0

Table 3
Selected population sizes and associated probabilities of success.

By adaptively scaling the variance, the results improve significantly. The probability of success on the parabolic ridge function and on Rosenbrocks function for instance become 1.0, even for the larger dimensionality. The only problem that cannot be solved with certainty within the bounds set is the Michalewicz problem. However, a vast improvement over the normal EDA can still be observed in the higher–dimensional case.

Although the results are a lot better for the EDA approach when the variance is scaled adaptively, the results of the CMA–ES are overall slightly better for the higher dimensionality (except for the parabolic ridge function and Michalewicz's function). The correlation–triggered adaptive–variance–scaling normal EDA approach has a slight advantage for the lower dimensionality. In

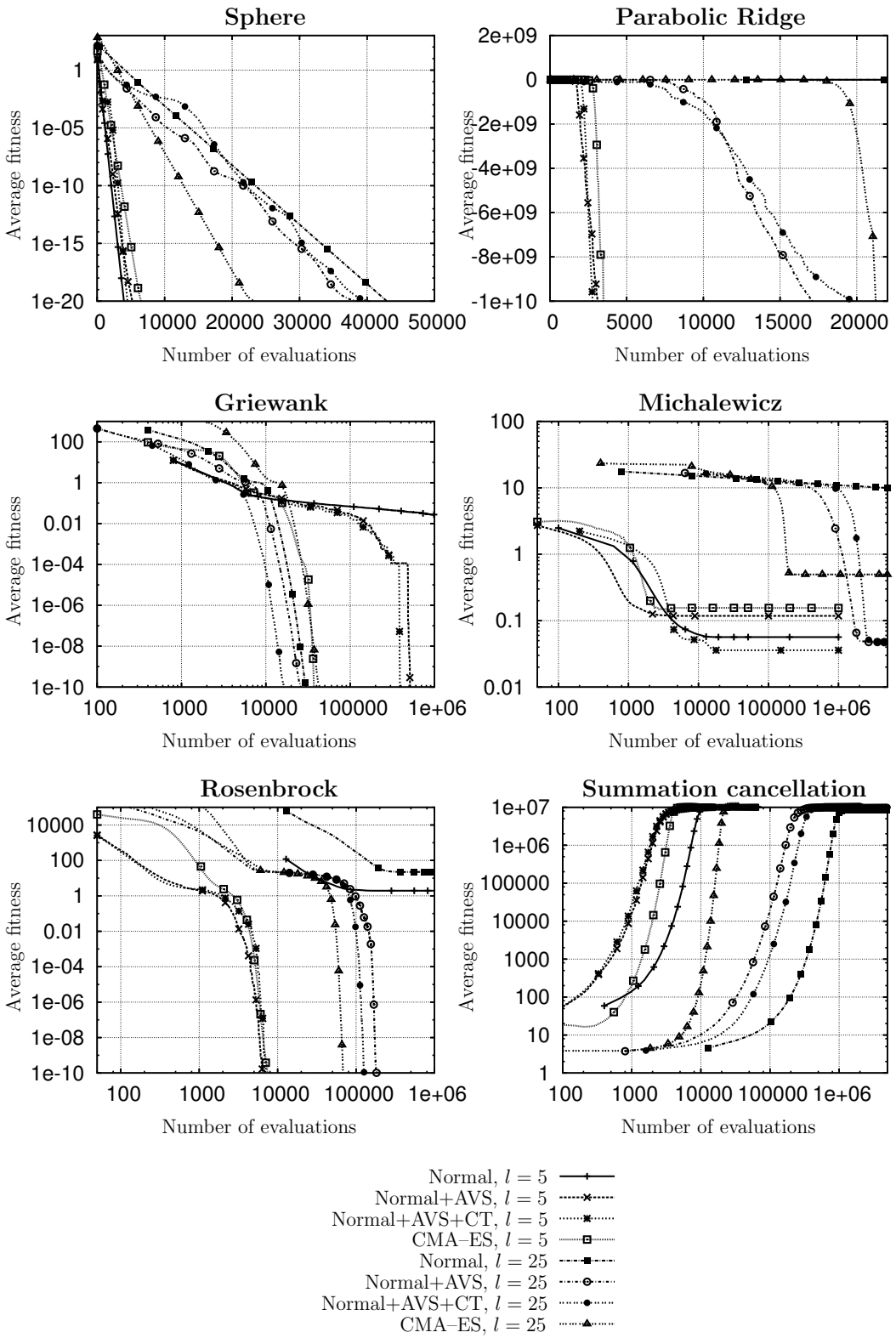

Fig. 7. Convergence behavior of various EDAs and of CMA–ES on all test problems.

addition to the performance with respect to number of required evaluations, the CMA–ES overall requires smaller population sizes as can be seen in Table 3. The reason for this is that in the CMA-ES, the probability distribution used to guide the search is not entirely rebuilt from scratch using only the data in the current set of selected solutions. Instead, the distribution is weighted over a path of generations past and hence represents an accumulation of information. Overall it can be concluded that the novel EDA approach is at least competitive with the CMA–ES.

5.3.3 Contribution of correlation trigger and of adaptive variance scaling

If the population size is at least as big as the population size that is required by the EDA without adaptive variance scaling, the correlation–triggering scheme indeed prevents the use of adaptive variance scaling as it detects that it is not required to efficiently find the optimum. This behavior is depicted in Figure 8. The results in Figure 7 and Table 3 however were selected on the basis of finding the smallest population size for which the problem was solved. Because the use of the correlation trigger allows for using smaller population sizes since adaptive variance scaling will then be triggered, the results in terms of number of evaluations required are still not as efficient as could be on simple functions such as the sphere function. On the other hand, on such problems that the EDA without adaptive variance scaling could already solve, the use of the correlation trigger allows for a wider range of population sizes to be used because it switches adaptive variance scaling on or off as required (i.e. depending on the size of the population). The added benefits of the correlation trigger do not come at the cost of degrading performance for other, more involved functions. The correlation trigger saves evaluations by forcing the variance scaling coefficient to 1 whenever possible, but doesn't fail to ensure adaptive variance scaling is used whenever required. For instance, on Rosenbrocks function, adaptive variance scaling is always required. Indeed, in Figure 8 it can be seen that the value for the variance scaling coefficient is most of the time kept at a value larger than 1, even if the population size is increased significantly. This conforms to the results in Table 3 where it can be seen that the EDA without variance scaling fails even with this large population size and hence, adaptive variance scaling is required. Overall we can conclude that our proposed remedy to the signaled deficiencies of using merely the normal distribution in an EDA significantly leverages the algorithm.

6 Summary and Outlook

In this article, we have analyzed the design of continuous EDAs starting from the lessons learned from research into discrete EDAs. The major lesson learned

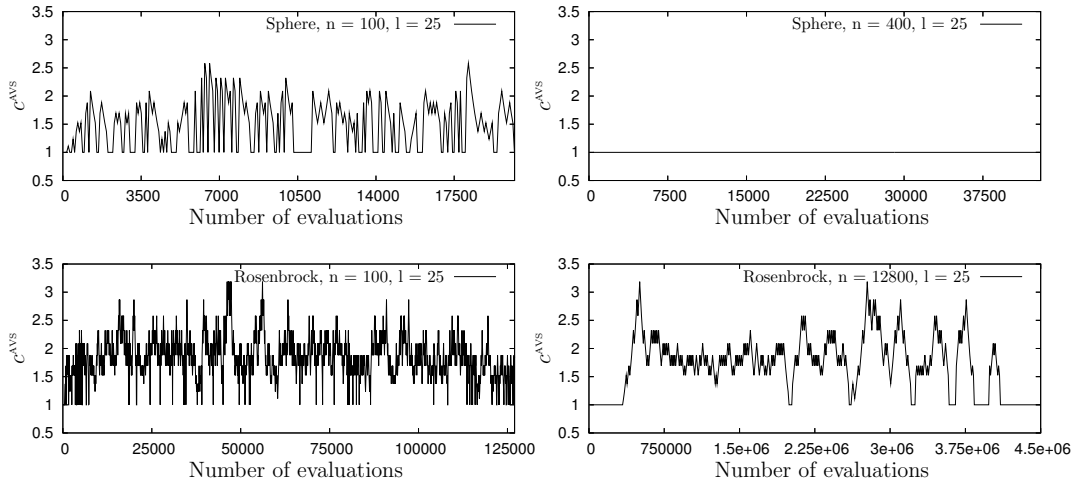

Fig. 8. Typical behavior of the correlation–triggered adaptive–variance–scaling coefficient c^{AVS} during a run on selected test problems using different population sizes.

that the inductive bias of the probabilistic model has to suit to the structure of the optimization problem at hand is valid in any domain. Problem decomposition allows to find efficient descriptions of the problem structure. In the discrete domain, this automatically maps to detecting dependency relations when estimating probability distributions as the basis of the inductive search bias of the optimization algorithm. This understanding then maps to typical GA concepts such as building blocks. In the continuous domain these concepts have no direct equivalent. Hence, before transferring the lessons learned from the discrete domain to the continuous domain, we must first generalize the lessons learned before we specialize them again for the domain at hand. In this article, we have made an important first step in this direction.

In continuous problems, the structure of a problem is characterized by the contours of the function to be optimized. Since the contours can take any shape and hence fitting the problem structure in the continuous case would require the intractable property of universal approximation, it is much more convenient to view problem structure as an arrangement of slopes and peaks in the search space. These simpler substructures are much easier to take into account and to build inductive search biases for.

Continuous EDAs rely on normal distributions. We have shown that the bias of the normal distribution does not fit well to slopes due to lack of generalization and as a consequence the EDA can get stuck. When searching around peaks, the bias of the normal distribution suffices to find the optimum. We argue that substructure identification is possible and beneficial in continuous EDA. Substructure identification relates to analyzing the local structure of the current search area so as to find out which shape dominates this search area. To accomplish proper substructure identification, we use the ranked correlation coefficient between the density of the approximated probability distribution

and the fitness of the set of selected solutions. On slopes, we then scale the variance of the EDA beyond its maximum–likelihood estimate. Experimental results on standard test problems indicate that the combination of these techniques greatly improves the search efficiency of continuous EDAs.

Concluding, the concept of problem structure needs to be carefully analyzed separately from its discrete counterpart when designing continuous EDAs. An assessment is needed of how the induced bias of a continuous EDA suits the characteristics of continuous optimization problems. Subsequently, techniques need to be developed to overcome potential drawbacks of using approximated density models that are unable to grasp in general the structure of the problem at hand. Our work is a first step into this direction. In future work, we plan to analyze in greater detail the possibilities to identify different substructures of the search space on the fly and to consequently design novel continuous EDAs. We believe that this is a promising area of research in continuous optimization with Estimation–of–Distribution Algorithms.

Acknowledgements

The authors would like to thank Stefan Minner and Franz Rothlauf for helpful suggestions and comments.

References

Ahn, C. W., Ramakrishna, R. S. and Goldberg, D. E. (2004). Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world, *in* K. Deb et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–2004*, Springer–Verlag, Berlin, pp. 840–851.

Anderson, T. W. (1958). *An Introduction to Multivariate Statistical Analysis*, John Wiley & Sons Inc., New York, New York.

Bengoetxea, E., Larrañaga, P., Bloch, I., Perchant, A. and Boeres, C. (2002). Learning and simulation of Bayesian networks applied to inexact graph matching, *Pattern Recognition* **35**(12): 2867–2880.

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

Blanco, R., Larrañaga, P., Inza, I. and Sierra, B. (2001). Selection of highly accurate genes for cancer classification by estimation of distribution algorithms, *in*

P. Lucas et al. (eds), *Proceedings of the Bayesian Models in Medicine Workshop at the 8th Artificial Intelligence in Medicine in Europe AIME–2001*, pp. 29–34.

Bosman, P. A. N. (2003). *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*, PhD thesis, University of Utrecht, Institute of Information and Computer Science.

Bosman, P. A. N. and Thierens, D. (1999). Linkage information processing in distribution estimation algorithms, *in* W. Banzhaf et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–1999*, Morgan Kaufmann, San Francisco, California, pp. 60–67.

Bosman, P. A. N. and Thierens, D. (2000). Expanding from discrete to continuous estimation of distribution algorithms: The IDEA, *in* M. Schoenauer et al. (eds), *Parallel Problem Solving from Nature – PPSN VI*, Springer–Verlag, Berlin, pp. 767–776.

Bosman, P. A. N. and Thierens, D. (2001a). Advancing continuous IDEAs with mixture distributions and factorization selection metrics, *in* M. Pelikan and K. Sastry (eds), *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference — GECCO–2001*, Morgan Kaufmann, San Francisco, California, pp. 208–212.

Bosman, P. A. N. and Thierens, D. (2001b). Exploiting gradient information in continuous iterated density estimation evolutionary algorithms, *in* B. Kröse et al. (eds), *Proceedings of the Thirteenth Belgium–Netherlands Artificial Intelligence Conference BNAIC–2001*, pp. 69–76.

Bosman, P. A. N. and Thierens, D. (2004). Learning probabilistic models for enhanced evolutionary computation, *in* Y. Jin (ed.), *Knowledge Incorporation in Evolutionary Computation*, Springer–Verlag, Berlin, pp. 147–176.

Cho, D.-Y. and Zhang, B.-T. (2001). Continuous estimation of distribution algorithms with probabilistic principal component analysis, *Proceedings of the 2001 Congress on Evolutionary Computation – CEC2001*, IEEE Press, Piscataway, New Jersey, pp. 521–526.

Cho, D.-Y. and Zhang, B.-T. (2002). Evolutionary optimization by distribution estimation with mixtures of factor analyzers, *Proceedings of the 2002 Congress on Evolutionary Computation – CEC2002*, IEEE Press, Piscataway, New Jersey, pp. 1396–1401.

Cho, D.-Y. and Zhang, B.-T. (2004). Evolutionary continuous optimization by distribution estimation with variational Bayesian independent component analyzers mixture model, *in* X. Yao et al. (eds), *Parallel Problem Solving from Nature – PPSN VIII*, Springer–Verlag, Berlin, pp. 212–221.

Ducheyne, E. I., De Wulf, R. R. and De Baets, B. (2002). Using linkage learning for forest management planning, *Late–Breaking Papers of the Genetic and Evolutionary Computation Conference — GECCO–2002*, pp. 109–114.

Etxeberria, R. and Larrañaga, P. (1999). Global optimization using Bayesian networks, *in* A. A. O. Rodriguez et al. (eds), *Proceedings of the Second Symposium on Artificial Intelligence CIMAF–1999*, Institute of Cybernetics, Mathematics and Physics, pp. 332–339.

Friedman, N. and Goldszmidt, M. (1996). Learning Bayesian networks with local structure, *in* E. Horvits and F. Jensen (eds), *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence UAI–1996*, Morgan Kaufmann, San Francisco, California, pp. 252–262.

Gallagher, M., Fream, M. and Downs, T. (1999). Real–valued evolutionary optimization using a flexible probability density estimator, *in* W. Banzhaf et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–1999*, Morgan Kaufmann, San Francisco, California, pp. 840–846.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Vol. 7 of *Series on Genetic Algorithms and Evolutionary Computation*, Kluwer Academic Publishers.

Grahl, J., Minner, S. and Rothlauf, F. (2005). Behaviour of UMDAc with truncation selection on monotonous functions, *Proceedings of the 2005 Congress on Evolutionary Computation – CEC2005*, IEEE Press, Piscataway, New Jersey, pp. 2553–2559.

Greene, W. H. (2003). *Econometric analysis*, Vol. 5, Prentice Hall, Upper Saddle River.

Hansen, N. (2006). The CMA evolution strategy: A comparing review, *in* J. A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea (eds), *Towards a new evolutionary computation. Advances in estimation of distribution algorithms.*, Springer–Verlag, Berlin.

Hansen, N., Muller, S. D. and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA–ES), *Evolutionary Computation* **11**(1): 1–18.

Harik, G. (1999). Linkage learning via probabilistic modeling in the ECGA, Technical Report 99010, IlliGAL, University of Illinois, Urbana, Illinois.

Harik, G. and Goldberg, D. E. (1996). Learning linkage, *Foundations of Genetic Algorithms 4*, pp. 247–262.

Harik, G. and Goldberg, D. E. (2000). Linkage learning through probabilistic expression, *Computer methods in applied mechanics and engineering* **186**: 295–310.

Kern, S., Müller, S. D., Hansen, N., Büche, D., Ocenasek, J. and Koumoutsakos, P. (2004). Learning probability distributions in continuous evolutionary algorithms — a comparative review, *Natural Computing* **3**(1): 77–112.

Larrañaga, P. and Lozano, J. (eds) (2001). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Vol. 2 of *Genetic Algorithms and Evolutionary Computation*, Kluwer Academic Publishers.

Larrañaga, P., Etxeberria, R., Lozano, J. A. and Peña, J. M. (2000). Optimization in continuous domains by learning and simulation of Gaussian networks, *in* M. Pelikan et al. (eds), *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference — GECCO–2000*, Morgan Kaufmann, San Francisco, California, pp. 201–204.

Lauritzen, S. L. (1996). *Graphical Models*, Clarendon Press, Oxford.

Mühlenbein, H. and Mahnig, T. (1999). FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions, *Evolutionary Computation* **7**(4): 353–376.

Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters, *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, pp. 178–187.

Ocenasek, J. and Schwarz, J. (2002). Estimation of distribution algorithm for mixed continuous-discrete optimization problems, *2nd Euro-International Symposium on Computational Intelligence*, pp. 227–232.

Ocenasek, J., Kern, S., Hansen, N. and Koumoutsakos, P. (2004). A mixed Bayesian optimization algorithm with variance adaptation, *in* X. Yao et al. (eds), *Parallel Problem Solving from Nature – PPSN VIII*, Springer–Verlag, Berlin, pp. 352–361.

Paul, T. and Iba, H. (2003a). Real–coded estimation of distribution algorithm, *Proceedings of the 5th Metaheuristics International Conference 2003 — MIC–2003*, pp. 60–1–60–6.

Paul, T. and Iba, H. (2003b). Reinforcement learning estimation of distribution algorithm, *in* E. Cantú-Paz et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–2003*, Springer–Verlag, Berlin, pp. 1259–1270.

Pelikan, M. and Goldberg, D. E. (2001). Escaping hierarchical traps with competent genetic algorithms, *in* L. Spector et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–2001*, Morgan Kaufmann, San Francisco, California, pp. 511–518.

Pelikan, M. and Goldberg, D. E. (2003). Hierarchical BOA solves ising spin glasses and maxsat, *in* E. Cantú-Paz et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–2003*, Springer–Verlag, Berlin, pp. 1271–1282.

Pelikan, M. and Mühlenbein, H. (1998). Marginal distribution in evolutionary algorithms, *Proceedings of the International Conference on Genetic Algorithms Mendel '98*, Brno, Czech Republic, pp. 90–95.

Pelikan, M., Goldberg, D. E. and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm, *in* W. Banzhaf et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–1999*, Morgan Kauffman, San Francisco, California, pp. 525–532.

Pelikan, M., Goldberg, D. E. and Lobo, F. (2002). A survey of optimization by building and using probabilistic models, *Computational Optimization and Applications* **21**(1): 5–20. Also IlliGAL Report No. 99018.

Rastegar, R. and Meybodi, M. R. (2005). A study on the global convergence time complexity of estimation of distribution algorithms, *in* D. Slezak et al. (eds), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 10th International Conference RSFDGrC–2005*, Springer–Verlag, Berlin, pp. 441–450.

Rothlauf, F. (2002). *Representations for Genetic and Evolutionary Algorithms*, number 104 in *Studies on Fuzziness and Soft Computing*, Springer, Berlin.

Rudlof, S. and Köppen, M. (1996). Stochastic hill climbing by vectors of normal distributions, *Proceedings of the First Online Workshop on Soft Computing (WSC1)*. Nagoya, Japan.

Schwefel, H. P. (1995). *Evolution and Optimum Seeking*, Wiley and Sons, New York.

Sebag, M. and Ducoulombier, A. (1998). Extending population–based incremental learning to continuous search spaces, *in* A. E. Eiben et al. (eds), *Parallel Problem Solving from Nature – PPSN V*, Springer–Verlag, Berlin, pp. 418–427.

Servet, I., Trave-Massuyes, L. and Stern, D. (1997). Telephone network traffic overloading diagnosis and evolutionary computation technique, *in* J. K. Hao et al. (eds), *Proceedings of Artificial Evolution '97*, Springer–Verlag, Berlin, pp. 137–144.

Shin, S.-Y. and Zhang, B.-T. (2001). Bayesian evolutionary algorithms for continuous function optimization, *Proceedings of the 2001 Congress on Evolutionary Computation – CEC2001*, IEEE Press, Piscataway, New Jersey, pp. 508–515.

Shin, S.-Y., Cho, D.-Y., and Zhang, B.-T. (2001). Function optimization with latent variable models, *in* A. Ochoa et al. (eds), *Proceedings of the Third International Symposium on Adaptive Systems ISAS–2001 — Evolutionary Computation and Probabilistic Graphical Models*, Institute of Cybernetics, Mathematics and Physics, pp. 145–152.

Sierra, B., Lazkano, E., Inza, I., Merino, M., Larrañaga, P. and Quiroga, J. (2001). Prototype selection and feature subset selection by estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips., *in* A. L. Rector et al. (eds), *Proceedings of the 8th Artificial Intelligence in Medicine in Europe AIME–2001*, Springer–Verlag, Berlin, pp. 20–29.

Tatsuoka, M. M. (1971). *Multivariate Analysis: Techniques for Educational and Psychological Research*, John Wiley & Sons Inc., New York, New York.

Thierens, D. (1999). Scalability problems of simple genetic algorithms, *Evolutionary Computation* **7**(4): 331–352.

Yuan, B. and Gallagher, M. (2005). On the importance of diversity maintenance in estimation of distribution algorithms, *in* H.-G. Beyer et al. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference — GECCO–2005*, Vol. 1, ACM Press, Washington DC, USA, pp. 719–726.