Probabilistic Modeling for Continuous EDA with Boltzmann Selection and Kullback-Leibeler Divergence

Cai Yunpeng
caiyp78@gmail.com

Sun Xiaomin
sxm123@tsinghua.edu.cn

Jia Peifa
dcs-jpf@tsinghua.edu.cn

State Key Lab of Intelligent Technology and Systems,
Tsinghua University, Beijing, P.R.China

ABSTRACT

This paper extends the Boltzmann Selection, a method in EDA with theoretical importance, from discrete domain to the continuous one. The difficulty of estimating the exact Boltzmann distribution in continuous state space is circumvented by adopting the multivariate Gaussian model, which is popular in continuous EDA, to approximate only the final sampling distribution. With the minimum Kullback-Leibeler divergence principle, both the mean vector and the covariance matrix of the Gaussian model can be calibrated to preserve the features of Boltzmann selection reflecting desired selection pressure. A method is proposed to adapt the selection pressure based on measuring the successfulness of the past evolution process. These works established a formal basis that helps to build probabilistic models in continuous EDA algorithms with adaptive parameters. The framework is incorporated in both the continuous UMDA and the EMNA algorithm, and tested in several benchmark problems. The experiment results are compared with some existing EDA versions and the benefit of the proposed approach is discussed.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization*

General Terms

Theory, Algorithms

Keywords

Evolutionary Computation, Estimation of Distribution Algorithms, Continuous Optimization, Boltzmann Selection

1. INTRODUCTION

Estimation of Distribution Algorithms (EDA) [19, 14], also called probabilistic model-building genetic algorithms (PMBGA) [24], iterated density estimation algorithms (IDEA) [7], or probabilistic modeling evolutionary algorithms [9], was proposed originally for combinatory optimizations. However, researchers have attempted to extend the approach to continuous optimization and have made many progresses. Current continuous EDA approaches involves extensions of discrete EDA from several aspects: space partitioning [25, 20, 28], introducing Gaussian distributions [26, 27, 13, 7, 20], or adopting pattern analyze methods [29]. A survey of EDA an their applications to continuous optimization can be found in [24, 14, 23] and [13, 7, 10].

Most continuous EDAs adopt Gaussian mutation to enable local search. It had been discovered that the model parameters, i.e., the variances/covariances, impact the performance greatly [5]. The determination of the variance or the covariance matrix is relatively independent to the algorithms. Existing continuous EDAs usually define the variance in the following manners:
(1) Using a constant or decaying variance [29, 10];
(2) Self-adaption: adjust the variance as in a (1,λ) evolutionary strategy [27, 21];
(3) Selecting-and-accounting: learn the variance from elite samples [27, 13, 15];
or the combination of (2) and (3) [21, 5]. The first method is too rigid to fit various problem landscapes. The latter two approaches do achieve substantial success, but there are also negative results concerning that self-adaption trends towards local search [27] and selecting-and-accounting might shrink the variance too fast [21]. The optimal choice of the variance remains an open problem.

In this paper we connect the problem of parameter learning to the framework of Boltzmann selection [18], which is a theoretically important method in combinatory optimization. We circumvent the trouble of estimating the exact Boltzmann distribution by adopting a Gaussian model to approximate the final sampling distribution, preserving the key features of the ideal distribution with the aid of the minimum Kullback-Leibler divergence [8] principle, thus get to a continuous version of Boltzmann selection which interprets parameter learning as deciding a proper selection pressure. We also developed a method to adapt the selection pressure based on measuring the successfulness of the last evolution cycle. Combining both results we achieve an adaptive framework of continuous EDA, called BGEDA, which tracks a temperature-decreasing Boltzmann distribution and performs well in the experiments.

In section 2, we describe the formal basis of continuous Boltzmann selection. In section 3, we give the method of

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

adapting selection pressure according to the progress in the last generation. In section 4 we connect our approach to some previous results in related fields. Experiment results are given in section 5 before concluding in section 6.

2. BOLTZMANN SELECTION IN CONTINUOUS OPTIMIZATION

In this section, we introduce some theoretical backgrounds and notations, then present our approach based on them.

2.1 Probabilistic Modeling

EDA features in iteratively creating a distribution model $\boldsymbol{q}(\boldsymbol{x})$ from selected individuals of current population and generating new population by sampling the model. For continuous EDAs, proposed distribution models include histogram [25], multivariate Gaussian distribution [15], Gaussian Networks [13] and mix of Gaussians [9, 20, 6], Helmholtz machines [29], etc. In this paper, we consider the following multivariate Gaussian distribution:

$$\boldsymbol{q}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Gamma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Gamma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \quad \boldsymbol{x} \in R^n \tag{1}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Gamma}$ is the covariance matrix.

Consider the following real-parameter optimization problem:

$$f(\boldsymbol{x}) : R^n \to R, \quad \boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\Omega}\{f(\boldsymbol{x})\}, \quad \Omega \subset R^n.$$

where $f(\boldsymbol{x})$, bounded on Ω, is usually called the fitness function in evolutionary algorithms. One question arises for the above probabilistic modeling approach:

Problem 1. Given a set of samples $\{\boldsymbol{X}_1, \boldsymbol{X}_2, .., \boldsymbol{X}_m | \boldsymbol{X}_i \in \Omega\}$ (the population) and their fitness $\{f(\boldsymbol{X}_1), f(\boldsymbol{X}_2), .., f(\boldsymbol{X}_m)\}$, what is the proper parameter set for $\boldsymbol{q}(\boldsymbol{x})$ to design a good optimization algorithm over $f(\boldsymbol{x})$?

2.2 Boltzmann Distribution and Boltzmann Selection

Inspired by statistical physics, many researchers [18, 2, 3, 4, 17, 10] have proposed to represent the object function $f(\boldsymbol{x})$ with a Boltzmann distribution:

$$\boldsymbol{p}(\boldsymbol{x}, T) = \frac{1}{Z(T)} e^{f(\boldsymbol{x})/T}, where\ Z(T) = \int_\Omega e^{f(\boldsymbol{x})/T} \boldsymbol{d}x. \tag{2}$$

Here $Z(T)$ is called the partition function, and $T > 0$ is the temperature parameter. In this way each value of the object function is mapped onto a sample probability. When $T \to \infty$ the distribution converges to a uniform distribution over the entire domain; on the other hand, when $T \to 0$ it converges to a distribution which uniformly charges only the optimal solutions. Some researchers also adopt the Gibbs distribution, which has almost the same property.

To simplify the notation, we denote $\beta = 1/T$, which is called the inverse temperature [16].

Mühlenbein pointed out that the Boltzmann distribution fulfills the Holland's Equation [17], which is an important property of a good population-based optimization algorithm. In a Boltzmann distribution with temperature t, for a schemata ξ with corresponding marginal distribution $\boldsymbol{x}_\xi$ and $\beta = 1/t$, it holds that

$$\frac{d\boldsymbol{p}(\boldsymbol{x}_\xi, t)}{d\beta} = \boldsymbol{p}(\boldsymbol{x}_\xi, t)(\hat{f}_\xi(\boldsymbol{x}, t) - \bar{f}(t)). \tag{3}$$

Here $\bar{f}(t)$ is the average fitness under current distribution and $\hat{f}_\xi(\boldsymbol{x}, t)$ is the average fitness of all solutions with schemata ξ. Equ. 3 describes that the probability of sampling a given schemata will increases in a rate proportional to its superiority over certain solutions as the temperature decreases. If an evolving population can track the temperature-decreasing Boltzmann distribution, it will improve iteratively and eventually converges to optimal solutions.

This conclusion gives an answer to Problem. 1 by arguing that a proper probabilistic model should approximate the Boltzmann distribution with the temperature decreasing towards zero. Thus the Boltzmann distribution has significant value in optimization theory.

The exact form of the Boltzmann distribution cannot be expressed without knowing the fitness function of every point in the space. To make use of the Boltzmann distribution, Mühlenbein et.al. proposed the Boltzmann Selection and the Boltzmann Estimation of Distribution Algorithm(BEDA) [18] for combinatory optimization. Given the estimated distribution $\boldsymbol{p}(\boldsymbol{x})$ and the temperature T, Boltzmann Selection calculates the following distribution

$$\boldsymbol{p}^s(\boldsymbol{x}) = \frac{\boldsymbol{p}(\boldsymbol{x})e^{f(\boldsymbol{x})/T}}{\sum_{\boldsymbol{y}} \boldsymbol{p}(\boldsymbol{y})e^{f(\boldsymbol{y})/T}} \tag{4}$$

and generates new samples according to it. To estimate $\boldsymbol{p}(\boldsymbol{x})$, BEDA, especially FDA [18], decomposes the distribution into marginal distributions and estimates them from sampled points. Such estimation is efficient in discrete cases where the state space is enumerable, but it is infeasible in continuous cases because there is no universal way to infer the form of the distribution based on finite points. So the Boltzmann selection cannot be applied to continuous optimization directly.

In this paper we attempt to implement Boltzmann selection without explicitly estimating $\boldsymbol{p}(\boldsymbol{x})$. We start from a population that approximately follows a Boltzmann distribution, and create a model to generate new samples that closely follows the distribution in Eq. 4. We adopt previous approach by Gallagher et.al. [10] which approximates the Boltzmann distribution of fixed temperature with a univariate Gaussian model via the Kullback-Leibler divergence, and extend it in three scopes: firstly, we adopt a multivariate Gaussian model; secondly, we change the calculation of K-L divergence so that both the mean vector and the covariance matrix can be decided; thirdly, we use a temperature-decreasing Boltzmann distribution instead of a fixed one.

2.3 Kullback-Leibler Divergence

The Kullback-Leibler divergence (KLD) [8], also known as the relative entropy, measures the distance between two distributions:

$$D(\boldsymbol{p}(\boldsymbol{x})||\boldsymbol{q}(\boldsymbol{x})) = \int_\Omega \boldsymbol{p}(\boldsymbol{x}) \log \frac{\boldsymbol{p}(\boldsymbol{x})}{\boldsymbol{q}(\mathbf{x})} \tag{5}$$

Eq. 5 is not symmetric to $\boldsymbol{p}(\boldsymbol{x})$ and $\boldsymbol{q}(\boldsymbol{x})$. If we try to approximate a target distribution $\boldsymbol{p}(\boldsymbol{x})$ with a model distribution $\boldsymbol{q}(\boldsymbol{x})$, we can either minimize $D(\boldsymbol{q}(\boldsymbol{x})||\boldsymbol{p}(\boldsymbol{x}))$ or $D(\boldsymbol{p}(\boldsymbol{x})||\boldsymbol{q}(\boldsymbol{x}))$. Previous works [3, 4, 10] choose to minimize $D(\boldsymbol{q}(\boldsymbol{x})||\boldsymbol{p}(\boldsymbol{x}))$ in order to avoid the trouble of calculating the partition function Z in Eq. 2. In this paper, on the contrary, we minimize $D(\boldsymbol{p}(\boldsymbol{x})||\boldsymbol{q}(\boldsymbol{x}))$ to acquire more information.

2.4 Continuous Boltzmann Selection with Gaussian Models

Now we treat the Boltzmann distribution $\boldsymbol{p}(\boldsymbol{x},T)$ in Eq. 2 as the target distribution, and $\boldsymbol{q}(\boldsymbol{x})$ in Eq. 1 the model function. If the random variable $\boldsymbol{x}$ follows the distribution $\boldsymbol{p}(\boldsymbol{x},T)$, it can be derived that

$$\begin{aligned}E[\ln \boldsymbol{q}(\boldsymbol{x})] &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)\ln \boldsymbol{q}(\boldsymbol{x})\boldsymbol{d}x \\ &= -H(\boldsymbol{p}(\boldsymbol{x},T)) - D(\boldsymbol{p}(\boldsymbol{x},T)||\boldsymbol{q}(\boldsymbol{x})).\end{aligned}$$

Here $E(\boldsymbol{x})$ is the expectation of $\boldsymbol{x}$ and $H(\boldsymbol{p})$ is the entropy of $\boldsymbol{p}$. From the above equation we see than minimizing $D(\boldsymbol{p}(\boldsymbol{x},T)||\boldsymbol{q}(\boldsymbol{x}))$ is equivalent to maximizing $E[ln\boldsymbol{q}(\boldsymbol{x})]$.

We write

$$\begin{aligned}E[\ln \boldsymbol{q}(\boldsymbol{x})] &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)\ln \boldsymbol{q}(\boldsymbol{x})\boldsymbol{d}x \\ &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Gamma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \\ &\qquad -\frac{1}{2}\ln|\boldsymbol{\Gamma}| - \frac{n\ln(2\pi)}{2}]\boldsymbol{d}x. \qquad (6)\end{aligned}$$

For a non-singular $\boldsymbol{\Gamma}$, denoting $\boldsymbol{B}=\boldsymbol{\Gamma}^{-1}$, the derivative of Eq. 6 is:

$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{\mu}} &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)\boldsymbol{B}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{d}x, \\ \frac{\partial E}{\partial \boldsymbol{B}} &= -\frac{1}{2}\int_\Omega \boldsymbol{p}(\boldsymbol{x},T)[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T - \boldsymbol{\Gamma}]\boldsymbol{d}x. \qquad (7)\end{aligned}$$

Forcing Eq. 7 equal to zero yields:

$$\begin{aligned}\boldsymbol{\mu} &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)\boldsymbol{x}dx = E(\boldsymbol{x}), \\ \boldsymbol{\Gamma} &= \int_\Omega \boldsymbol{p}(\boldsymbol{x},T)(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{d}x = Var(\boldsymbol{x}). \qquad (8)\end{aligned}$$

which is the parameter set that maximizes Eq. 6. If a random sampling can be drawn according to $\boldsymbol{p}(\boldsymbol{x},T)$, this result will be trivial since it is merely the maximum likelihood estimation of the mean vector and covariance matrix of a certain Gaussian distribution. However, it has been noted above that such a distribution cannot be acquired unless $T\to\infty$, which is a uniform random distribution. Boltzmann selection suggests a means to solve this problem.

Replacing $1/T$ with β, the Boltzmann selection fulfills the following lemma:

LEMMA 1. *If* $\boldsymbol{p}_1(\boldsymbol{x}) = \frac{e^{\beta f(\boldsymbol{x})}}{\int_{\boldsymbol{z}\in\Omega} e^{\beta f(\boldsymbol{z})}\boldsymbol{d}z}$ *is a Boltzmann distribution, then*

$$\boldsymbol{p}_2(\boldsymbol{x}) = \frac{\boldsymbol{p}_1(\boldsymbol{x})e^{\Delta\beta f(\boldsymbol{x})}}{\int_{\boldsymbol{z}\in\Omega}\boldsymbol{p}_1(\boldsymbol{z})e^{\Delta\beta f(\boldsymbol{z})}\boldsymbol{d}z} = \frac{e^{(\beta+\Delta\beta)f(\boldsymbol{x})}}{\int_{\boldsymbol{z}\in\Omega} e^{(\beta+\Delta\beta)f(\boldsymbol{z})}\boldsymbol{d}z}$$

is also a Boltzman distribution.

The proof of the lemma can be found in [18] with the summation replaced by the integration. From this lemma we see that Boltzmann selection generates a lower-temperature Boltzmann distribution from a higher one. A larger $\Delta\beta$ indicates a higher selection pressure, and vice versa.

LEMMA 2. *If* $\boldsymbol{X}_1,\boldsymbol{X}_2,..,\boldsymbol{X}_m$ *are i.i.d random variables* $\sim \boldsymbol{p}(\boldsymbol{x})$*, and* $g(\boldsymbol{x})$ *is a bounded function on* Ω*,* $E[g(\boldsymbol{x})] = \int_\Omega \boldsymbol{p}(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}x$*, then*

$$\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^m g(\boldsymbol{X}_i) \xrightarrow{P} E[g(\boldsymbol{x})].$$

PROOF. Since $g(\boldsymbol{x})$ is bounded, $\forall$ i, the variance

$$D[g(X_i)] < \infty.$$

Note that functions of independent random variables are also independent random variables, from the *Chebyshev*'s theorem we get that $\forall\,\varepsilon>0$,

$$\begin{aligned}&\lim_{m\to\infty}P(|\frac{1}{m}\sum_{i=1}^m g(\boldsymbol{X}_i) - E[g(\boldsymbol{x})]| < \varepsilon) \\ =\ &\lim_{m\to\infty}P(|\frac{1}{m}\sum_{i=1}^m g(\boldsymbol{X}_i) - \frac{1}{m}\sum_{i=1}^m E[g(\boldsymbol{X}_i)]| < \varepsilon) \\ =\ &1. \quad \square\end{aligned}$$

From the above lemma we get the following result:

THEOREM 1. *If* $\boldsymbol{X}_1,\boldsymbol{X}_2,..,\boldsymbol{X}_m$ *are i.i.d random variables following the Boltzmann distribution* $\boldsymbol{p}_1(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x},1/\beta)$ *for the bounded fitness function* $f(\boldsymbol{x})$*,* $\boldsymbol{p}_2(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x},1/(\beta+\Delta\beta))$ *is another Boltzmann distribution, then for any function* $g(\boldsymbol{x})$ *satisfying*

$$\int_\Omega \boldsymbol{p}_1(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}x < \infty, \int_\Omega \boldsymbol{p}_2(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}x < \infty,$$

$$\lim_{m\to\infty}\frac{\sum_{i=1}^m[e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}g(\boldsymbol{X}_i)]}{\sum_{i=1}^m e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}} \xrightarrow{P} \int_\Omega p_2(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}x.$$

PROOF.

$$\begin{aligned}&\lim_{m\to\infty}\frac{\sum_{i=1}^m[e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}g(\boldsymbol{X}_i)]}{\sum_{i=1}^m e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}} = \lim_{m\to\infty}\frac{\frac{1}{m}\sum_{i=1}^m[e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}g(\boldsymbol{X}_i)]}{\frac{1}{m}\sum_{i=1}^m e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}} \\ &= \frac{\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^m[e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}g(\boldsymbol{X}_i)]}{\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^m e^{\Delta\beta\cdot f(\boldsymbol{X}_i)}} \xrightarrow{P} \frac{\int_\Omega \boldsymbol{p}_1(\boldsymbol{x})e^{\Delta\beta\cdot f(\boldsymbol{x})}g(\boldsymbol{x})\boldsymbol{d}x}{\int_\Omega \boldsymbol{p}_1(\boldsymbol{x})e^{\Delta\beta\cdot f(\boldsymbol{x})}\boldsymbol{d}x} \\ &= \int_\Omega \frac{\boldsymbol{p}_1(\boldsymbol{x})e^{\Delta\beta\cdot f(\boldsymbol{x})}}{\int_\Omega \boldsymbol{p}_1(\boldsymbol{z})e^{\Delta\beta\cdot f(\boldsymbol{z})}\boldsymbol{d}z}g(\boldsymbol{x})\boldsymbol{d}x = \int_\Omega \boldsymbol{p}_2(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}x. \quad \square\end{aligned}$$

This theorem suggests a method to derive some sufficient statistics of one Boltzmann distribution with higher β from a group of samples drawn from another Boltzmann distribution with lower β, without knowing the exact form of either one. For a population $\{X_i\}\sim \boldsymbol{p}_1(\boldsymbol{x},1/\beta)$ and a desired distribution $\boldsymbol{p}_2(\boldsymbol{x},1/(\beta+\Delta\beta))$, applying the theorem to Eq. 8 yields

$$\boldsymbol{\mu} = \sum_{i=1}^{m} e^{\Delta\beta \cdot f(\boldsymbol{X}_i)} \boldsymbol{X}_i / \sum_{i=1}^{m} e^{\Delta\beta \cdot f(\boldsymbol{X}_i)},$$

$$\boldsymbol{\Gamma} = \frac{\sum_{i=1}^{m} [e^{\Delta\beta \cdot f(\boldsymbol{X}_i)} (\boldsymbol{X}_i - \boldsymbol{\mu})(\boldsymbol{X}_i - \boldsymbol{\mu})^T]}{\sum_{i=1}^{m} e^{\Delta\beta \cdot f(\boldsymbol{X}_i)}}. \quad (9)$$

Eq. 9 gives a Gaussian model that closely executes the Boltzmann selection in continuous space. Recalling the fact that the uniform random distribution is the Boltzmann distribution with $T \to \infty$, we can start from a random population, and iteratively create a Gaussian model from current population then sample from the model to achieve a new one. If we assume that the characters of Boltzmann distribution is still preserved in the new population, we get to a way of evolving the population following a temperature-decreasing Boltzmann distribution, thus a continuous version of Boltzmann selection, without estimating the current distribution $\boldsymbol{p}(\boldsymbol{x})$. The iteration resembles the BEDA framework mentioned in section 2.2, and we call our version the Boltzmann-Gaussian Estimation of Distribution Algorithm(BGEDA), which is described in Table 1. The problem of model parameter choosing in section 1 is also answered by connecting it to determining the selection pressure represented by $\Delta\beta$, which is called the annealing schedule in [16]. We will deal with it in section 3.

Compared to adopting Eq. 4 directly, the above approach is beneficial in that merely a rough estimation is needed to keep track of the target Boltzmann distribution, thus we do not need a large sample set to figure out the exact probability, which makes the framework feasible.

The replacement step in Table 1, which is popular in many current EDA versions, introduces additional disturbances to the distribution. However, since the Gaussian model favors points around the mean vector while replacement encourages points of good fitness, they seem to compensate each other in more cases than to magnify the errors, provided that the sample size is large enough.

We implemented two algorithms from the above framework. The first one is called BG-UMDA(Boltzmann-Gaussian Univariate Marginal Distribution Algorithm), which utilizes univariate Gaussian function to estimate the marginal distribution in each dimension, and sample each variable independently. The second one is called EMNA-B(Estimation of Multivariate Normal Algorithm with Boltzmann Selection), which adopt a single multivariate Gaussian model containing all variables. For the replacement step, we use a simple tournament replacement strategy [14].

3. DETERMINING THE ANNEALING SCHEDULE

In the last section a problem is raised on how to choose a proper selection pressure, thus a proper increment of the inverse temperature $\Delta\beta$. Previous schedules for combinatory optimization [16] can be adopted directly, but we try to interpret them with deeper statistical background. We reveal that the annealing schedule can be connected to implicit assumptions about the probabilistic distribution of the fitness, and give an approach to determine the annealing schedule based on these assumptions. The key idea is to measure the actual improvement of the population, thus the actual $\Delta\beta$ achieved, during the last cycle. A large $\Delta\beta$ is then afforded to a successful evolution record, and a small one is afforded to a failed one.

Table 1: Algorithm of Boltzmann-Gaussian Estimation of Distribution Algorithm (BGEDA)

1	Generate a population P according to the uniform random distribution; Initialize a random $\Delta\beta \geq 0$;
2	do {
3	With $\Delta\beta$, Calculate the parameters of the multivariate Gaussian model M with Eq.9 based on P;
4	Sample an offspring population Q from M;
5	Update the population P with Q using the Replacement Strategy;
6	Update $\Delta\beta$;
7	}until (stop criterion reached).

Consider the distribution of samples in the space of fitness: $y = f(\boldsymbol{x}), \quad y \in V \subset R$. Suppose we have created a model M for a given optimization algorithm. Sampling from M yields a random fitness value y with probability density function $p_Y(y)$. A population $P = \{y_1, y_2, .., y_m\}$ is generated from M with fitness $y_1, y_2, .., y_m$. The sample mean of the population is u and the variance is v^2.

Now we execute a Boltzmann selection to the current distribution with unknown parameter $\Delta\beta$:

$$p'_Y(y) = \frac{p_Y(y) e^{\Delta\beta y}}{Z(\Delta\beta)}, \quad Z(\Delta\beta) = \int_V p_Y(y) e^{\Delta\beta y} dy.$$

Sampling from $p'_Y(y)$ yields an offspring population $O = \{z_1, z_2, .., z_m\}$, with the mean fitness u'. Now given the two populations P and O, we can estimate the annealing temperature $\Delta\beta$ with the maximum likelihood method:

$$\Delta\hat{\beta} = \arg\max_{\Delta\beta} \{\ln(\prod_{i=1}^{m} p'_Y(z_i)\}. \quad (10)$$

By solving Eq. 10 we get the actual improvement of the last evolution step and we can simply adopt it as the attempted annealing temperature of the coming step. The problem is that $p_Y(y)$ is generally unknown. However, by assuming the form of $p_Y(y)$ we can derive solutions with some desired property. Consider the following two cases:

1) $p_Y(y) \sim N(u, v^2), \quad y \in (-\infty, \infty)$.

This assumption treats repeated sampling from M as a random work from P, with equal chance of getting better or worse results, which is by far the case of most optimization processes, if the bounded condition of $f(\boldsymbol{x})$ is ignored. With this distribution the solution of Eq.10 can be given as:

$$Z(\Delta\beta) = e^{u\Delta\beta + v^2 \Delta\beta^2/2}, \quad \Delta\hat{\beta} = \frac{u' - u}{v^2}. \quad (11)$$

This solution is almost identical to one of the annealing schedules proposed in [16], which is derived from the Taylor expansion of the average fitness. The difference is that in [16] u and v^2 are the mean and variance of current population and u' is the desired average fitness.

2) $p_Y(y) = \frac{1}{v} e^{-\frac{(u+v)-y}{v}}, \quad y \in (-\infty, u+v]$.

This assumption means that the fitness has an estimated upper bound $(u+v)$ and follows a exponential distribution, which is analogue to the case that the population converges near the optimum and it is hard to find a good sample.

Solving Eq.10 with this distribution yields:

$$Z(\Delta\beta) = \frac{e^{\Delta\beta(u+v)}}{1+v\Delta\beta}, \quad \Delta\hat{\beta} = \frac{1}{v}[\frac{u'-u}{v-(u'-u)}]. \tag{12}$$

The second term in the solution of $\Delta\beta$ in Eq. 12 usually varies slower than the first one, since a larger deviation in current population usually indicates more chances to improve the average fitness. Thus this solution, in some sense, resemble the Standard Deviation Schedule (SDS) in [16], with a slow-varying coefficient replacing a tunable constant.

Adopting this method, the new $\Delta\beta$ in step 6 in Table. 1 can be calculated from the old and the new population P of each generation. We set $\Delta\beta = 0$ for the first generation, since it is observed in the experiments in section 5 that the initial value of $\Delta\beta$ will not impact the result much. Also in these experiments we did not find significant benefits to adopt SDS or the annealing schedule in Eq. 12, so we use the schedule suggested by Eq. 11.

4. RELATED WORKS

Berny [2, 3, 4] and Gallagher [9, 10] had investigated the approach of approximating the Boltzmann distribution of fixed temperature by the means of gradient descending, respectively. For univariate Gaussian distribution, they got the following update rule:

Berny:

$$\Delta\mu = \alpha(x-\mu)(f(x) + T(1+\log Q(x))). \tag{13}$$

Gallagher's standard continuous PBIL:

$$\Delta\mu = \alpha(X_i - \mu)\arg\max_i\{f(X_i)\}. \tag{14}$$

Gallagher's PBIL-KLD:

$$\Delta\mu = \alpha(X_i - \mu)\sum_i [f(X_i) - \bar{f}], \tag{15}$$

where $\alpha \propto (n\sigma^2 T)^{-1}$.

The distinction between these works and our approach had been discussed in section 2.2 and 2.3. Now we investigate their relationship. Rewriting Eq. 7 in the univariate form, and approximate it with Theorem 1, we derive

$$\frac{\partial E}{\partial \mu} = \frac{\sum_i e^{\Delta\beta f(X_i)}(X_i - \mu)}{\sigma^2 \sum_i e^{\Delta\beta f(X_i)}}.$$

With the approximation $\Delta\beta f(\boldsymbol{x}) \ll 1$, and $\sum_i (X_i - \mu) \approx 0$, we get

$$\frac{\partial E}{\partial \mu} \approx \frac{\Delta\beta}{n\sigma^2}(X_i - \mu)\sum_i [f(X_i) - \bar{f}].$$

We see that Eq. 15 is the special case of the Boltzmann selection when the selection pressure is very low. With the same method we can get that Eq. 14 is also a special case of Boltzmann selection with $\Delta\beta \to \infty$, which represent a high selection pressure. This explains the experiment results in [10] that PBIL-KLD converges slower than standard continuous PBIL in smooth fitness landscapes, but performs better in noisy and fragmentary landscapes. Eq. 13 is analogue to Eq. 15 but the Gibbs distribution is used instead of the Boltzmann distribution.

Berny also investigated the multivariate case and gave the gradient equation of both the mean vector and the covariance matrix to approximate a fixed temperature Boltzmann distribution, but it is exhausting in computation.

Sebag and Ducoulombier [27] proposed the PBILc algorithm with the following update rule:

$$\Delta\mu = \alpha(X_{best1} + X_{best2} - X_{best} - \mu), \tag{16}$$

which is inspired from differential evolution. They also provides four means of handling the Gaussian variance:
(1) Use a constant variance;
(2) Adapt the variance as in a (1,λ) ES;
(3) Estimate the variance from the diversity of the elite samples, i.e., selecting-and-accounting;
(4) Selecting-and-accounting with incremental learning.

Larrañaga et.al. proposed a serial of continuous EDA which are called UMDAc, MIMICc, EGNA and EMNA [13, 14, 15] respectively. These algorithms utilize univariate Gaussian marginal distribution, multivariate Gaussian distribution, or Gaussian networks to create models, respectively. The major difference between UMDAc, EMNA and our algorithms proposed in section 2.4 is that they adopt selecting-and-accounting scheme to determine the model parameter, and handle the selection pressure by the proportion of selected samples. In section 5 we will compare our approach with these algorithms.

It has been studied [12, 11] that for UMDAc with truncate selection or 2-to-1 tournament selection and in monotonous landscape, the optimum might be missed if it is far away from the initial model center, which shows the limit of the selecting-and-accounting approach. Whether our approach overcomes this problem is not clear, but with the weighted average over the entire population the initial model center is less likely to be severely biased and the variances will decrease more softly.

5. EXPERIMENTS

In this section we test the proposed algorithms in section 2.4 with some well-known benchmark problems for continuous EDA and compare their performance with previous reports. We follow the exact problem definitions, stop criteria and evaluating criteria of these benchmarks as they were first introduced in the area of EDA. The expression of the problems are not given due to the length limit of the paper.

5.1 Algorithms Tested

Since neither BG-UMDA nor EGNA-B adopts local structures, we compare them with continuous EDA versions that are also uni-modal. We divide the candidate algorithms with their learning style and their method of parameter choosing:

Algorithms with no dependency learning:

- Standard Continuous PBIL(sc-PBIL), with fixed variance parameter;
- BG-UMDA, which adopts continuous Boltzmann Selection;
- UMDAc, which adopt selecting-and-accounting scheme;

Table 2: Results for Test Problem 1

Algorithms	F1	F2	F3
sc-PBIL	4.43 ± 0.4	7.54 ± 0.36	18.7 ± 0.63
(10,50)-ES	2.91 ± 0.45	7.56 ± 1.52	399.07 ± 6.97
PBILc	4.76 ± 0.78	10.99 ± 1	4803 ± 4986
BG-UMDA	4.83 ± 0.57	11.32 ± 0.72	$10^7 \pm 0.0002$
EMNA-B	337 ± 110	326 ± 79	5.80 ± 0.99

- PBILc, which learn the variance incrementally;
- (μ,λ)-ES, which adopts self-adapt variance.

Algorithms with dependency learning:

- EMNA-B, which adopts continuous Boltzmann Selection;
- EMNA[1], which adopt selecting-and-accounting scheme;
- EBNA, which adopt selecting-and-accounting scheme and learns the network structure.

We also attempted to compare PBIL-KLD with these algorithms, but there is not much previous results reported, and it seems to converge slowly in the proposed benchmarks to the best of our effort, so we did not give the results on it.

5.2 Test Problems and Results

The first set of benchmark problems are from [27], which had been proposed for testing the performance of PBILc. Each problem has a global maximum 10^7. Five algorithms are compared in this experiment. For BG-UMDA, the population is set to 50 for $F1$, 75 for $F2$ and 100 for $F3$. For EMNA-B, a fixed population of 1500 is used. For sc-PBIL, as the base algorithm, we strive to seek a optimal parameter set for it by combining various population (25-200), learning rate(0.01-0.2) and variance(0.1%-10% to the range of the variable). The results of PBILc and ES are taken from [1].

The average best fitness of 20 runs for each algorithm after 200000 evaluations is depicted in Table 2. From the result we see that BG-UMDA and PBILc outperform ES and sc-PBIL in all these problems. The two algorithms performs almost the same in $F1$ and $F2$, but in $F3$ where there is no dependency between variables, BG-UMDA significantly performs better. On the other hand, EMNA-B performs significantly better than other algorithms in problems with variable dependencies ($F1$ and $F2$), but performs very poor in $F3$. This is not surprising since in EMNA-B there are too many parameters(over 5000) to learn compared to the number of evaluations but none of them is useful to speed up the search.

The second experiment continuous to compare PBILc, BG-UMDA and EMNA-B with the test problems suggested in [13], which are all of 10 dimension. Among which the $SumCan$ function has a global maximum of 10^5 and other functions have global minima of 0. The average best fitness in 100 runs after 300000 evaluations is measured.

For EMNA-B, the population size is fixed as 400. For BG-UMDA, the population size is chosen as 700,700,100. For PBILc, we search for a combination of parameters with the population from 50 to 2000, and the learning rate from 0.01 to 0.2. The result is depicted in Table 3. Reports for UMDAc, EGNA and ES taken from [13] are also listed.

Table 3: Results for Test Problem 2

Algorithms	SumCan	Schwefel	Griewangk
PBILc	91002 ± 28611	unstable[2]	0.11 ± 0.57
ES	5910	0[3]	0.034477
UMDAc	53460	0.13754	0.011076
EGNA	100000	0.0250	0.008175
BG-UMDA	79682 ± 17960	0.0090 ± 0.0030	0.0010 ± 0.0056
EMNA-B	100000 ± 0	$2.7e{-}31 \pm 1.0e{-}31$	$5.8e{-}05 \pm 5.8e{-}4$

From Table. 3 we see that EMNA-B performs almost the best in all problems. BG-UMDA outperforms UMDAc in all three functions and outperform PBILc and EGNA in two of the three functions.

In the third experiment we try to evaluate the performance of both proposed algorithm in a more extensive way. The set of benchmark problems in [14] is chosen.[4] The benchmark involve five 10-dimension and five 50-dimension problems. The stop criterion is met when the best fitness is within 10^{-6} error of the global optimal, or the number of evaluations reaches 300000, or the algorithm converges. We used a population size of 400 for BG-UMDA in all problems, also for EMNA-B in all 10-dimension problems, and 1000 for EMNA-B in 50-dimension problems. The average best fitness in 10 runs is evaluated. Table 4 depicts the performance data of BG-UMDA and EMNA-B, and Figure 1 displays the comparison of their performance to the baseline algorithms. In the figure all algorithms are aligned with mean fitness error to the global optimum (in logarithm coordination) and the mean number of evaluations so that the algorithm with the best performance is place at the left-bottom corner in each subplot. For the runs that reach the target precision, the errors are all set to 10^{-6}.

To summarize the results in Figure 1. BG-UMDA and EMNA-B altogether achieve best performance in 8 problems out of 10. The deviation data in Tab. 4 validates that these improvements are statistically significant. Each algorithm significantly outperform the baseline algorithms in 6 cases. They are dominated only in the Rosenbrock functions by ES. This result shows the good quality of our approach.

In these three experiments, generally our approach performs better than the algorithms using other schemes of determine model parameters: constant, self-adaption and selecting-and accounting. This may attribute to that with Boltzmann selection the entire population is considered and a balanced variance is achieved. But the ES with self-adaption seems to be stronger in local search, and PBILc sometimes performs better since it takes long-term and negative (Eq. 16) feedbacks. Another benefit of our approach is that the only hand-tuned parameter is the population, which makes it easy to apply. The experiment results are at least a validation to the quality of our approach as adaptive algorithms.

[1] In fact EMNA and EBNA are a class of algorithms, in the experiments we pick out their best results regardless of the species.

[2] The errors vary from 10^3 to 10^{-8}, thus the average fitness is meaningless.

[3] The precision is not given.

[4] The range of the Ackley function is not declared in the book, we use the one suggested in [22], i.e., [-32.768,32.768].

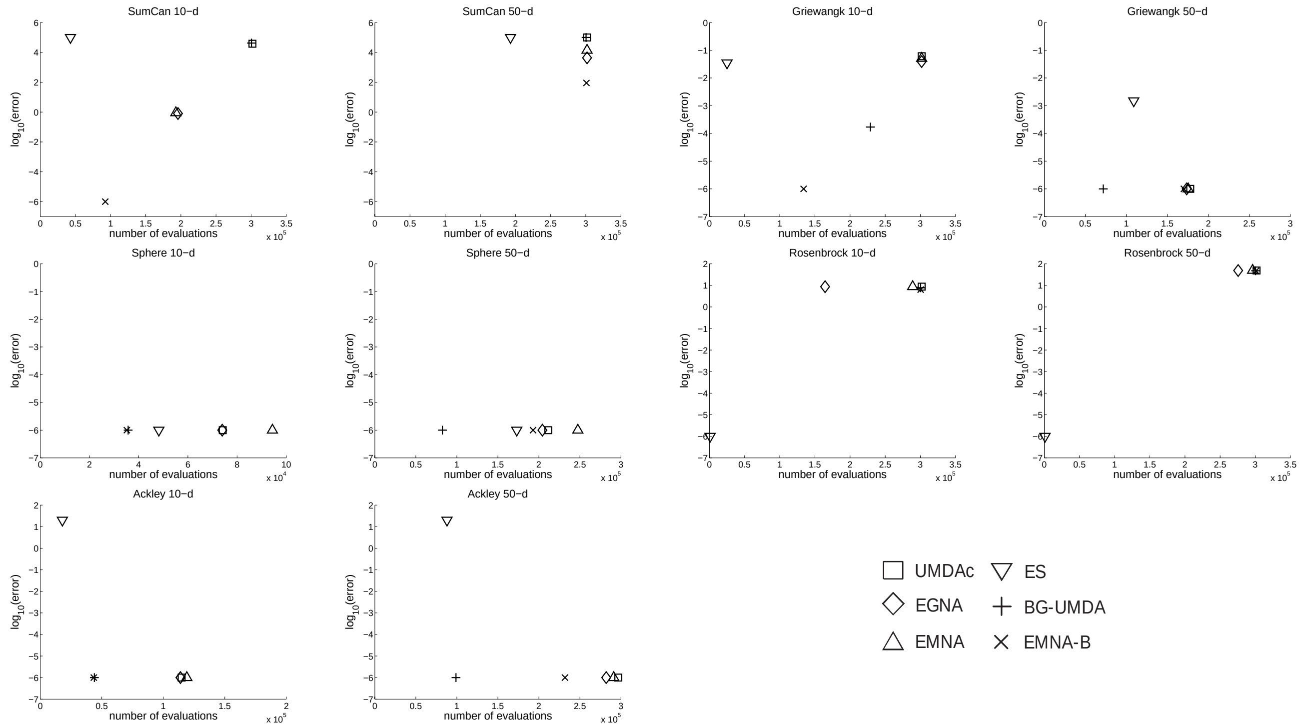

Figure 1: Performance Comparison on Test Problem 3

Table 4: Performance Data of BG-UMDA and EMNA-B on Test Problem 3

BG-UMDA		
Function	Fitness	No. Eval.
SumCan-10d	$5.8e+4 \pm 2.3e+4$	300400 ± 0
SumCan-50d	1.39 ± 0.10	300400 ± 0
Griewangk-10d	$1.27e-4 \pm 4.0e-4$	229000 ± 64000
Griewangk-50d	$8.8e-7 \pm 7e-8$	71880 ± 420
Sphere-10d	$5.9e-7 \pm 1.8e-7$	35720 ± 840
Sphere-50d	$8.4e-7 \pm 8e-8$	82400 ± 460
Rosenbrock-10d	7.74 ± 0.08	300400 ± 0
Rosenbrock-50d	47.54 ± 0.07	300400 ± 0
Ackley-10d	$8.3e-7 \pm 1.6e-7$	44000 ± 530
Ackley-50d	$9.6e-7 \pm 4e-8$	98920 ± 530
EMNA-B		
Function	Fitness	No. Eval.
SumCan-10d	$100000 \pm 1.1e-7$	92520 ± 840
SumCan-50d	99910 ± 160	301000 ± 0
Griewangk-10d	$7.4e-7 \pm 1.1e-7$	134000 ± 47000
Griewangk-50d	$9.2e-7 \pm 5e-8$	170100 ± 1700
Sphere-10d	$7.5e-7 \pm 2.1e-7$	35200 ± 420
Sphere-50d	$8.8e-7 \pm 1.1e-7$	192900 ± 1600
Rosenbrock-10d	6.33 ± 0.37	300400 ± 0
Rosenbrock-50d	47.08 ± 0.44	301000 ± 0
Ackley-10d	$8.4e-7 \pm 1.0e-7$	43560 ± 610
Ackley-50d	$9.42e-7 \pm 4e-8$	231800 ± 4300

6. CONCLUSION AND FUTURE WORKS

In this paper the Boltzmann selection in combinatory EDA is extended to continuous optimization problems by adopting multivariate Gaussian models and the K-L divergence. The approach manages to approximate the Boltzmann distribution without demanding a large sample set and leads to a method of choosing model parameters in continuous EDAs to inherits the evolving property of Boltzmann selection. An annealing schedule is also given to adaptively determine a proper selection pressure based on evaluating the improvement of the average fitness in the last evolution cycle. These works lead to a framework of continuous EDA which is nearly parameter-free. Experiment results validate the quality of the approach in a set of benchmark problems.

It has been pointed out in [30] that adopting a variance larger than the maximum-likelihood estimation will be sometimes necessary for local search. Merging this observation with the Boltzmann selection framework in this paper might help to understand the nature of adapting variance strategies [21, 5], which will be a tough work.

Although the proposed framework is implemented in unimodal EDAs in this paper, it can be directly extended to multi-modal EDAs that adopts clustering by simply applying it to separated clusters. For EDAs with structural learning, if there is some necessity to incorporate Boltzman selection into them, theoretical works must be done to synthesize the framework in this paper with machine learning theory concerning specified model structure.

7. REFERENCES

[1] S. Baluja. Population Based Incremental Learning - A Method for Integrating Genetic Search Based

Function Optimisation and Competitive Learning. Technical Report CMU-CS-94-163, CMU, Jun. 1994.

[2] A. Berny. An Adaptive Scheme for Real Function Optimization Acting As a Selection Operator. In *First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pages 140–149. IEEE, 2000.

[3] A. Berny. Selection and Reinforcement Learning for Combinatorial Optimization. In *Parallel Problem Solving from Nature - PPSN VI*, pages 601–610. Springer Verlag, 2000.

[4] A. Berny. Statistical Machine Learning and Combinatorial Optimization. In *Theoretical Aspects of Evolutionary Computation*, pages 287–306. Springer Verlag, 2001.

[5] P. Bosman and J. Grahl. Matching Inductive Search Bias and Problem Structure in Continuous Estimation of Distribution Algorithms. Technical Report 03/2005, Dept. of Logistics, Mannheim Business School, 2005.

[6] P. Bosman and D. Thierens. Mixed IDEAs. Technical Report UU-CS-2000-45, Utrech University, 1999.

[7] P. Bosman and D. Thierens. Expanding from Discrete to Continuous Estimation of Distribution Algorithms: The IDEA. In *Parallel Problem Solving from Nature - PPSN VI*, pages 767–776, Paris, France, Sep. 2000. Springer-Verlag.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[9] M. Gallagher. *Multi-layer Perceptron Error Surfaces: Visualization, Structure and Modelling*. PhD thesis, University of Queensland, 2002.

[10] M. Gallagher and M. Frean. Population-Based Continuous Optimization, Probabilistic Modelling and Mean Shift. *Evolutionary Computation*, 13(1):29–42, 2005.

[11] C. González, J. Lozano, and P. Larrañaga. Mathematical Modelling of UMDAc Algorithm with Tournament Selection: Behaviour on Linear and Quadratic Functions. *International Journal of Approximate Reasoning*, 31(2):313–340, 2002.

[12] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with Truncation Selection on Monotonous Functions. In *The 2005 IEEE Congress on Evolutionary Computation*, pages 2553–2559, Edinburgh, Scotland, 2005. IEEE.

[13] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks. In *Proceedings of the GECCO-2000 Workshop in Optimization by Building and Using Probabilistic Models*, pages 201–204, San Francisco, CA, USA, 2001. Morgan Kaufmann.

[14] P. Larrañaga and J. Lozano. *Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[15] P. Larrañaga, J. A. Lozano, and E. Bengoetxea. Estimation of Distribution Algorithms based on Multivariate Normal and Gaussian Networks. Technical Report EHU-KZAA-IK-1-01, University of the Basque Country, 2001.

[16] H. Mühlenbein and T. Mahnig. A New Adaptive Boltzmann Selection Schedule SDS. In *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, pages 183–190. IEEE, 2001.

[17] H. Mühlenbein and T. Mahnig. Evolutionary algorithms and the Boltzmann Distribution. In *Foundations of genetic algorithms (FOGA-7)*, pages 133–150. Morgan Kaufmann, 2003.

[18] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez. Schemata, Distributions and Graphical Models in Evolutionary Optimization. *Journal of Heuristics*, 5:215–247, 1999.

[19] H. Mühlenbein. The Equation for Response to Selection and Its Use for Prediction. *Evolutionary Computation*, 5(3):303–346, 1997.

[20] J. Oceanasek and J. Schwarz. Estimation of Distribution Algorithm for Mixed Continuous-discrete Optimization Problems. In *2nd Euro-International Symposium on Computational Intelligence*, pages 227–232, Kosice, Slowakia, 2002. IOS Press.

[21] J. Ocenasek, S. Kern, N. Hansen, S. Müller, and P. Koumoutsakos. A Mixed Bayesian Optimization Algorithm with Variance Adaptation. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 352–361. Springer Verlag, 2004.

[22] T. Paul and H. Iba. Real-Coded Estimation of Distribution Algorithm. In *Proceedings of The Fifth Metaheuristics International Conference*, 2003.

[23] M. Pelikan. *Bayesian Optimization Algorithm: From Single Level to Hierarchy*. PhD thesis, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL. USA, 2002. IlliGAL Report No. 2002023.

[24] M. Pelikan, D. Goldberg, and F. Lobo. A Survey to Optimization by Building and Using Probabilistic Models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[25] M. Pelikan, D. Goldberg, and S. Tsutsui. Getting the Best of Both Worlds: Discrete and Continuous Genetic and Evolutionary Algorithms in Concert. *Information Sciences*, 156(3-4):147–171, 2003.

[26] S. Rudolf and M. Köppen. Stochastic Hill Climbing by Vectors of Normal Distributions. In *Proceedings of the First Online Workshop on Soft Computing*, Nagoya, Japan, 1996.

[27] M. Sebag and A. Ducoulombier. Extending Population-Based Incremental Learning to Continuous Search Spaces. In *Parallel Problem Solving from Nature - PPSN V.*, pages 418–427, Amsterdam, Netherlands, Sep. 1998. Springer-Verlag.

[28] I. Servet, L. Trave-Massuyes, and D. Stern. Telephone Network Traffic Overloading Diagnosis and Evolutionary Techniques. In *Proceedings of the Third European Conference on Artificial Evolution, (AE'97)*, pages 137–144. Springer Verlag, 1997.

[29] S. Shin and B.-T. Zhang. Bayesian Evolutionary Algorithms for Continuous Function Optimization. In *Proc. 2001 Congress on Evolutionary Computation*, pages 508–515. IEEE, 2001.

[30] B. Yuan and M. Gallagher. On the Importance of Diversity Maintenance in Estimation of Distribution Algorithms. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 719–726. ACM Press, 2005.