Diversity loss in general estimation of distribution algorithms

Jonathan L. Shapiro

University of Manchester, Manchester, UK. M13 9PL
jls@cs.man.ac.uk

Abstract. A very general class of EDAs is defined, on which universal results on the rate of diversity loss can be derived. This EDA class, denoted SML-EDA, requires two restrictions: 1) in each generation, the new probability model is build using only data sampled from the current probability model; and 2) maximum likelihood is used to set model parameters. This class is very general; it includes simple forms of many well-known EDAs, e.g. BOA, MIMIC, FDA, UMDA, etc. To study the diversity loss in SML-EDAs, the trace of the empirical covariance matrix is the proposed statistic. Two simple results are derived. Let N be the number of data vectors evaluated in each generation. It is shown that on a flat landscape, the expected value of the statistic decreases by a factor $1-1/N$ in each generation. This result is used to show that for the Needle problem, the algorithm will with a high probability never find the optimum unless the population size grows exponentially in the number of search variables.

1 Introduction

Estimation of distribution algorithms (EDAs) are search algorithms inspired by evolutionary algorithms. Whereas evolutionary algorithms use a population of configurations to search for a solution to an optimization problem, EDAs use a probability function instead. This probability function models the population which it replaces. For this reason, EDAs are also often called "probability-model building evolutionary algorithms". A range of EDAs have been proposed and developed, both for continuous and discrete search spaces, and a number of successful applications have been reported. A recent book [1] reviews the field.

One of the appeals of EDAs is that the probability models can represent and *learn* the structure between the search variables. The earliest EDAs treated each variable independently [2, 3]. Later EDAs allowed a structured relationship between the variables. This allows correlations between the variables to be maintained during search. Since the effectiveness of genetic algorithms, and most other heuristic search algorithms, is highly dependent on the move operators, the fact that EDAs can learn move operators is a very enticing feature.

Despite the appeal of EDAs and the reported successes, there are difficulties in applying them effectively. One difficulty, which is the primary issue of this paper, is that under certain circumstances certain EDAs can get into states from where they cannot find the optimum no matter how long they are run. This is because they have lost their

diversity; it is analogous to fixation in a mutation-free genetic algorithm. Thus, to apply EDAs effectively, the control parameters of the algorithm must be chosen to avoid this situation. However, this may be difficult to do. For independent-variable EDAs, it has been shown [4, 5] that the appropriate settings of these control parameters is very different for different problems. For example, the learning rate in PBIL needs to be sufficiently small to insure that the optimum is found, and it must be exponentially small in the system size in some problems, but need only scale as low-order polynomial of the system size in others. This makes it very difficult to set this control parameter in advance. Similar results hold for UMDA, where the population size must grow as a problem-dependent function of the number variables; exponentially for some problems, polynomially in others.

It has not be known whether these results hold for EDAs with more complex variable structure. A number of different EDAs have been proposed, which are distinguished by the structure imposed on the variables and by the method used to learn the probability model. This makes theoretical analysis difficult, first because there are so many different models to analyze, and second, because the analysis of models which change their structure at each generation is difficult. Indeed, much of the theoretical work has focused on the simplest, independent-variable EDAs (e.g. [6, 7, 5]). An example of theoretic work on population sizing in a specific non-independent EDAs is [8].

It is worth emphasizing that although it is expected that the runtime of algorithms will depend very strongly on the problem, effective algorithms should work with fairly generic settings of the control parameters. Otherwise, one will have to dedicate substantial computational resources to searching control-parameter space, resources which could be applied to searching for the optimum using a more robust algorithm.

In this paper, two rigorous results are derived which hold for an entire class of EDAs. It is a fairly unrestricted class, including EDAs which learn structure, such as the Bayesian Optimization Algorithm (BOA) [9] in its simplest form, as well as simple EDAs such as UMDA [3] which is one of the earliest independent-variable EDAs, and many others. The first result concerns the expected diversity loss per generation when searching on a flat landscape, and the corresponding expected time to completely lose diversity (fixation). The second result is a lower bound on the minimum population size required to insure that the optimum is found when searching for a particular configuration on an otherwise flat landscape (the so-called *needle in a haystack* problem). Both results are universal for the entire class of EDAs.

2 Estimation of Distribution Algorithms

I will consider the search space to consist of L variables, all of which take values from some finite set $\mathcal{A}$, i.e. $\mathbf{x} = (x_1, x_2, \ldots, x_L) \in \mathcal{A}^L$. The goal is to find the configuration which maximize an objective function $f : \mathcal{A}^L \to \mathcal{R}$.

At the heart of the EDA is a class of probability functions from which a probability function is chosen at each generation. In general, this has two parts: the *structure* defines which variables interact with which, and the *parameters* define the form of the interaction. The structure is discrete and denoted $\mathcal{S}$; the parameters are continuous and denoted $\mathcal{P}$. To avoid confusion, the term *parameter* will always be used to refer to a

parameter of the probability model; the term *control-parameter* will be used to refer to parameters of the algorithm, such as the population size.

To express this mathematically, the assumption of EDAs is that for each variable x_i, there is a set of variables, called its parents, on which it depends. Let $\pi(i)$ denote the set of parents of component i. The underlying assumption is that the joint probability of all the variables $P(\mathbf{x})$, factorizes.

$$P(\mathbf{x}) = \prod_{i=1}^{L} P(x_i|\mathbf{x}_{\pi(\mathbf{i})}). \tag{1}$$

Here, we use the shorthand $\mathbf{x}_{\pi(\mathbf{i})}$ to denote the vector consisting of just those components which are parents of i.

The set of parents for each variable is what constitutes the structure of the probability model. Once the structure has been determined, one needs to estimate values for each of the factors in equation (1). These values are what constitutes the parameters of the probability model.

The generic EDA is roughly as follows. Start with a random population of M vectors. Then implement a loop consisting of

1. Select N vectors using a selection method.
2. Learn the probability model $(\mathcal{S}, \mathcal{P})$ from the selected population.
3. Sample M vectors from the probability model.

There are many variations. Selection is often done by applying truncation selection to the sampled population. Alternatively, the selected vectors replace a fraction of the selected population from the previous generation rather than the entire population. Likewise, the probability model can be built from scratch at each generation, or or can use the model from the previous generation(s) to build the current model. Another source of variation in EDAs is in the allowed structure: UMDA treats each variable independently. MIMIC assumes a chain of interactions, so every variable except the root and the terminal variables are the parent of one node and the child of another. BOA uses a general directed acyclic graph to represent the structure. And there are many others.

2.1 SML-EDA: A restricted class of EDAs

In this work, we consider a restricted class of EDAs for which we will derive some simple results. We will need three assumptions.

Assumption 1. The probability distribution in each generation is built using only data which was sampled from the probability model of the previous generation.

Assumption 2. The parameters of the estimated model are chosen using maximum likelihood.

Assumption 3. The sample size M and the size of the population used to build the model N are of a constant ratio independent of the number of variables L.

Assumption 1 means that data in generation t can only affect data in generation $t+1$ through the probability model. This rules out taking data directly from previous generations to put into the current population. Thus, there can be no elitism or mixing of the population with populations from previous generations.

The results of this work apply irregardless of the mechanism used to generate the structure of the probability model. However, once the structure is chosen, Assumption 2 requires the parameters of the model are those which maximize the probability of the data given the probability model,

$$\mathcal{P}_{\text{ML}} = \text{argmax}_{\mathcal{P}} \, \text{prob}\,(\text{population}|(\mathcal{S}, \mathcal{P})). \tag{2}$$

This estimation is widely used, because it is easily computable from empirical frequencies. For example, once the structure of the model is determined, a typical model parameter will correspond to the probability that a particular component takes the particular value, given the value of its parents. If assumption 2 holds, this probability is estimated by the following ratio,

$$P(x_i|\mathbf{x}_{\pi(\mathbf{i})}) \approx \frac{N(x_i, \mathbf{x}_{\pi(\mathbf{i})})}{N(\mathbf{x}_{\pi(\mathbf{i})})}, \tag{3}$$

where $N(x, y)$ denotes the number of times in the data that x and y takes their values.

In this paper, we will explore two results which hold for all EDAs for which these two assumptions hold. For the purpose of this paper, we will refer to all such EDAs as SML-EDAs, to denote Simple, Maximum-Likelihood EDAs.

Definition 1. *The class of EDAs for which assumptions 1 and 2 hold are called SML-EDAs.*

This class includes a wide group of EDAs and can include EDAs which assume independent structure, EDAs which have non-trivial fixed structure, and EDAs which learn their structure from the data. Certainly many of the standard EDAs, such as UMDA, MIMIC, FDA, BOA, fall in this class in their simplest form.

There are two ways in which EDAs typically fail to be in the class SML-EDA. First, because they use data from several previous generations to generate the probability model. The second reason is that the parameters are not set using maximum likelihood. PBIL, for example would not be in this class because its values are estimated as a linear combination of the current values and the maximum likelihood ones. Mühlenbein and Mahnig [10] suggest setting parameters for FDA using a maximum posteriori method rather than maximum likelihood; if that is done the resulting EDA is not in this class.

Finally, in this work we are interested in asymptotic behavior for large L. Assumption 3 ensures that a single control parameter governs the population size. Under these three assumptions, the EDA works like this:

1. Initialize `sample_pop` to be a random population of size M;
2. **Repeat**
   (a) Produce `select_pop` by selecting N from `sample_pop`;
   (b) Learn the structure of the probability model $\mathcal{S}$ from `select_pop` by any means;

(c) Learn the parameters of the probability model $\mathcal{P}$ from `select_pop` using equation (3).
(d) Update `sample_pop` by sampling M vectors independently from the probability model defined by $(\mathcal{S}, \mathcal{P})$.

3. **Until** some stopping criterion met

3 Diversity loss in SML-EDAs on a flat fitness landscape

The first result concerns the rate of diversity loss in SML-EDAs on a flat landscape. To measure the diversity in a population of size N, we will use the *trace of empirical co-variance matrix*. Let ν_i^A be the empirical frequency at which the component i takes the value A, i.e.

$$\nu_i^A = \frac{1}{N} \sum_{\mu} \delta(x_i^\mu = A), \tag{4}$$

where x_i^μ is component i of population member μ, and δ is an indicator function; 1 if its argument is true, and 0 if its argument is false. The diversity measure we will use is,

$$v = \sum_i \frac{1}{|\mathcal{A}|} \sum_a \nu_i^a \left(1 - \nu_i^a\right), \tag{5}$$

where $|\mathcal{A}|$ is the number of values component x_i can take, which is assumed to be the same for all components.

Whenever the value of a component fixates, i.e. is the same in all members of a population, the corresponding $\sum_a \nu_i^a \left(1 - \nu_i^a\right)$ will be zero. So, complete fixation implies $v = 0$. The random population has the maximum value of v. Finally this is related to the covariance matrix which is defined as the expectation that two components both take the value A, minus the expectation that they take this value independently, summed over all values,

$$C_{ij} = \frac{1}{|\mathcal{A}|} \sum_a \left[\langle \delta(x_i = a)\delta(x_j = a)\rangle - \langle \delta(x_i = a)\rangle \langle \delta(x_j = a)\rangle\right] \tag{6}$$

where angled brackets $\langle\cdot\rangle$ denotes expectation. The quantity v_t is the trace of the empirical estimate of C at generation t.

Using this measure of diversity, it is trivial to derive the diversity loss on a flat landscape, on which all vectors have the same fitness. On a flat landscape, sampling M values followed by selection of a population of size N is equivalent to sampling a population of size N.

Theorem 1. *For EDAs in class SML-EDA on a flat landscape, the expected value of v is reduced in each generation by a factor of 1 minus the inverse population size,*

$$\langle v_t \rangle = \langle v_{t-1} \rangle \left(1 - \frac{1}{N}\right). \tag{7}$$

A detailed proof will be given elsewhere. To get from generation $t-1$ to t, there are two steps. First the probability distribution is created from the data. Second, a new population is created from the probability model. The proof is almost trivial. Starting from the value of v_{t-1}, the model is built. Since the parameters are set by maximum likelihood, the marginals are equal to the frequencies for each component. Then sampling is done. Since the landscape is flat, we can combine sampling with selecting, and replace that step with the single step of sampling N data vectors from the probability model. It is well known that empirical variance is reduced by a factor $(1 - 1/N)$ from the parent population. Thus $\langle v_t \rangle = v_{t-1} \left(1 - \frac{1}{N}\right)$. Finally, we average over the right-hand side to get the result.

This provides a prediction for the expected diversity loss which is universal for all EDA in the class SML-EDA,

$$\langle v_t \rangle = v_0 \left(1 - 1/N\right)^t , \tag{8}$$

which decays with characteristic time approximately equal to the population size for large N. Assuming a random initial population, $v_0 = L/|\mathcal{A}|(1 - 1/|\mathcal{A}|)$.

4 A universal bound for the minimum population size in the Needle problem

Next we consider an SML-EDA searching for a particular configuration on an otherwise flat landscape. This problem is sometimes called the *needle in the haystack* problem, or the *Needle* problem. I.e. there is one special state (the "needle") which has a high fitness value, and all other configurations have the same low fitness value.

Using the result from the previous section, it is possible to derive a lower bound for the minimum population size N needed to solve this problem. Let T_N be the time that the needle is first sampled. The shorthand $T_N = \infty$ will mean that the needle is never sampled. (Time is measured in units of iterations of the algorithm. One time-step refers to one cycle of selection, model building, and sampling.) We will assume, as above, that the search space consists of L variables, each of which takes $|\mathcal{A}|$ possible values. Asymptotics will be for large L; $|\mathcal{A}|$ is assumed to be a constant.

To derive a lower bound on the minimum population size required to ensure that the optimum is found, a bound can be derived for the probability that the needle is *never* sampled,

$$\mathrm{prob}\left(T_N = \infty\right) \geq B(N, L). \tag{9}$$

Then, the follow holds.

Theorem 2. *In the limit that $L \to \infty$ such that $N^2 L |\mathcal{A}|^{-L} \to 0$,*

$$B(N, L) = 1 - O\left(|\mathcal{A}|^{-L/2}\right)$$

for any EDA in SML-EDA searching on the Needle problem.

This result shows that any SML-EDA will almost never find the needle if the population size is $o(\sqrt{|\mathcal{A}|^L / L})$. I.e. the population size must grow at least as fast as $\sqrt{|\mathcal{A}|^L / L}$ for the optimum to be found.

Theorem 2 will be proved in two steps. First, a time t^* is defined so that, if the algorithm has run for that length of time without finding the needle, it is highly likely that it never will find the needle. Next it is shown that if the population size grows (with L) sufficiently slowly, it is highly likely that the algorithm will run for t^* steps without finding the needle.

The first step towards proving Theorem 2 relies on the following result.

Lemma 1. *Let t^* be defined as*

$$t^* = -\frac{\frac{L}{2}\log\left(|\mathcal{A}|\right) + \log\left(LNv_0\right)}{\log\left(1 - 1/N\right)}. \tag{10}$$

If the needle has not been found after a time $t \geq t^*$, the probability that the needle will never be found is greater than $1 - \epsilon$, where $\epsilon = |\mathcal{A}|^{-L/2}$. Mathematically, this is expressed,

$$\text{prob}\left(T_N = \infty | T_N > t^*\right) \geq 1 - |\mathcal{A}|^{-L/2}. \tag{11}$$

Proof. To compute $\text{prob}\left(T_N = \infty | T_N > t^*\right)$ first observe that at time t, the expected variance is given by equation (8) and will be very small. Because the variance is positive, the fact that the expected variance is small means that the actual variance must also be small with a high probability. The largest the actual variance can be with a probability greater than or equal to $1 - \epsilon$ is $\langle v \rangle / \epsilon$, for any ϵ between 0 and 1 (see, for example, [11]). In other words,

$$\text{prob}\left[v(t) \leq \frac{\langle v(r) \rangle}{\epsilon}\right] \geq 1 - \epsilon. \tag{12}$$

The idea is to choose t^* to be large enough so that

$$v_t \leq \frac{\langle v_t \rangle}{\epsilon} \leq \frac{1}{N}\left(1 - \frac{1}{N}\right) \tag{13}$$

The reason is that if v_t is so small, there must be fixation at every component except possibly one component. (Fixation of a component i means that the variable x_i takes the same value in all vectors in the population.) Since maximum likelihood is used to build the probability model, once fixation happens at any component, that component will remain fixed for the rest of the run of the algorithm. If $L - 1$ components are fixed, the needle will only be sampled if they are fixed at values found in the needle. The probability of this is no more than $|\mathcal{A}|^{-(L-1)}$. Thus, we can write

$$\text{prob}\left(T_N = \infty | T_N > t^*, v_t \leq 1/N - 1/N^2\right) > 1 - |\mathcal{A}|^{-(L-1)}. \tag{14}$$

This is the probability that the needle is never found assuming that the needle has not been found up to time greater than t^*, and given that v_t is small enough that we know that $L - 1$ components are fixed.

We are not certain that v_t appropriately small, it is just probable so. However, the probability that v_t is small as we assume is ϵ. Take

$$\epsilon = |\mathcal{A}|^{-L/2}. \tag{15}$$

Then it is the leading order term and

$$\text{prob}\left(T_N = \infty | T_N > t^*\right) = 1 - O(\epsilon). \tag{16}$$

It only remains to compute t^*. This is done by setting equation (13) to be equality, and solving equation (8) for t. The solution is equation (10).

□

The next step in proving Theorem 2 is to consider the probability of not finding the needle during the t^* steps.

Lemma 2. *Let t^* be defined as in equation (10). The probability that the needle is not found after t^* steps obeys*

$$\text{prob}\left(T_N > t^*\right) \geq 1 - \frac{N^2}{|\mathcal{A}|^L}\left[\frac{L}{2}\log\left(|\mathcal{A}|\right) + \log\left(LN\right)\right]. \tag{17}$$

Proof. Choose t^* as in equation 10. Since random search is optimal for this problem[1], the probability that SML-EDA does not find the needle in time t^* obeys

$$\text{prob}\left(T_N > t^*\right) \geq \left(1 - |\mathcal{A}|^{-L}\right)^{t^* N}. \tag{18}$$

(Remembering that N vectors are considered at each time-step.)

The result is found by putting into this equation the value for t^*, and using the convexity of $\log$ and $\exp$ and other standard inequalities to simplify the expression, □

Proof of Theorem 2

Proof. The probability of never finding the needle can be decomposed into,

$$\text{prob}\left(T_N = \infty\right) = \text{prob}\left(T_N = \infty | T_N > t^*\right)\text{prob}\left(T_N > t^*\right). \tag{19}$$

with t^* defined as previously. Lemma 1 gives a lower bound for the first factor on the right side of equation (19). Lemma 2 gives a lower bound for the second factor. Combining them gives the following,

$$\text{prob}\left(T_N = \infty\right) \geq 1 - |\mathcal{A}|^{-(L/2)} - N^2|\mathcal{A}|^{-L}\left[\frac{L}{2}\log\left(|\mathcal{A}|\right) + \log\left(LN\right)\right] \tag{20}$$

$$+ \text{ higher order terms in } |\mathcal{A}|^{-L}. \tag{21}$$

If in the limit that $L \to \infty$, N grows sufficiently slowly that the third term vanishes, then the probability of never finding the needle will go to 1. Thus, if

$$N = o\left(\frac{|\mathcal{A}|^{L/2}}{\sqrt{L}}\right), \tag{22}$$

the leading term in equation 20 will be $1 - |\mathcal{A}|^{-L/2}$ and as $L \to \infty$ the needle will never be found. □

[1] among algorithms which make no attempt to prevent visiting the same configuration multiple times

5 The expected runtime for the Needle problem and the limits of universality

The content of Section 4 is basically if the algorithm runs for time t^* without finding the needle, the algorithm has likely fixated and will never find the needle. Since number vectors processed is tN which must be of order $|\mathcal{A}|^L$ for the needle to be found, N must be approximately the size of $|\mathcal{A}|^L/t^*$ for the needle to be found. This result is *universal*, it holds for the entire class SML-EDA.

The next task is to show that when the population size is sufficiently large, the probability of finding the optimum approaches one, and to produce an estimate of the run time when N is large enough so that the needle is typically found. It is not clear that this can be done for the entire class SML-EDA. Lemma 1 gives a universal upper bound on the search time *given the needle is found*.

Corollary 1. *If the needle is found, the time to find it is bounded above by t^* with probability $1 - |\mathcal{A}|^{-L/2}$.*

However, this is not very informative. If the population size is smaller than the critical value, as given in equation 22, this bound is not useful, since the needle will almost never be found. If the population grows much faster than the critical value, it is likely that this bound is not tight. For example, if the population size $N = O(|\mathcal{A}|^L)$ it is likely that the optimum will be found in the first few generations. If the population size obeys the critical scaling, $N^2 = O(|\mathcal{A}|^L/L)$, Corollary 1 suggests that the algorithm is efficient; the runtime is asymptotically the same as random search. However, the results herein are not sufficient to show that the probability of finding the optimum is needle is close to one in this case.

The reason that it is not possible to investigate the regime in which the optimum is find using the methods of this paper, is that the particular statistic, v_t gives one-sided information. When it is small, there is definitely fixation, independent of the structure of the probability model. However, when it is near its initial value, that does *not imply that there is no fixation* in models with non-trivial structure. As an example, consider a chain model with binary variables. Each variable x_i can take the values 0 or 1, and the parent of variable i is node $i-1$. Variable 1 is a root and has no parent. In other words, the assumed probability model is $P(\mathbf{x}) = P(x_1)\prod_{i=2}^{L} P(x_i|x_{i-1})$. Suppose it fixates such that $P(x_1) = 1/2$, $P(x_i = 1|x_{i-1} = 0) = 1$ and $P(x_i = 0|x_{i-1} = 1) = 1$. The only vectors which can be generated are $01010\ldots$ and $101010\ldots$. This fixation would be totally invisible to the statistic v_t which would continue to equal its initial value.

One conclusion is that convergence conditions will not be universal, but will be particular to the EDA. (This paper is essentially about non-convergence conditions.) A particular statistic, sensitive to the type of probability model might be necessary. For example, the statistic used here is appropriate for SML-UMDA, for which it can be shown that if the population size N and runtime T obey $TN = O(|\mathcal{A}|^L)$ and $T/N = O(|\mathcal{A}|^{-L\delta})$ for $\delta > 0$, the algorithm behaves asymptotically like random search on the Needle problem and is therefore efficient. For other EDAs other statistics may be required to study algorithmic efficiency. Alternatively, it is possible that the decay of some general property of the covariance matrix, e.g. its rank, may be used to show convergence results for the general class SML-EDA, but that remains to be seen.

6 Conclusions

With inappropriate settings, many EDAs can reach a state from which the probability of ever finding the optimum is zero. This is due to diversity loss which cannot be restored. If any component of the data vectors does not take one of its allowed values anywhere in the entire population, that value can never be restored. If that value is required in the optimum, the optimum will never be sampled. The flat landscape is the simplest problem in which this can be studied. We have shown that this diversity loss is the same for a whole class of EDAs. A consequence of this is that for a problem which is almost everywhere flat, such as the Needle problem, the probability of diversity loss before the optimum is sampled is also universal for the class, and we have shown that it requires an exponentially large population size to avoid this.

It is important to go beyond these results. In many other search problems, the landscape will not be flat, but there will be many directions which are essentially flat. It was shown in UMDA[5] and PBIL [4] that the rate of diversity loss relative to the rate of search in non-flat directions helped to understand how control parameters needed to be set to ensure a reasonable probability of finding the optimum. Presumably the same will be true in arbitrary EDAs. However, unlike diversity loss, I expect that search in the non-flat dimensions not to be universal, but to depend on the structure of the probability model. This remains to be investigated.

References

1. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
2. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competive learning. Technical Report CMU-CS-94-163, Computer Science Department, Carnegie Mellon University (1994)
3. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions i: Binary parameters. In: Proceedings of PPSN IV. (1996) 178–187
4. Shapiro, J.L.: The sensitivity of pbil to its learning rate and how detailed balance can remove it. In Jong, K.A.D., Poli, R., Rowe, J.E., eds.: Foundations of Genetic Algorithms 7. Morgan Kaufmann (2003) 115–132
5. Shapiro, J.L.: The detailed balance principle in estimation of distribution algorhithm. Evolutionary Computation **13**(1) (2005) 99–124
6. Droste, S.: Not all linear functions are equally difficult for the compact genetic algorithm. In: GECCO'05, ACM (2005) 679–686
7. Gonzalez, C., Lozano, J., Larrañaga, P.: Mathematical modelling of umcac algorithm with tournament selection: Behaviour on linear and quadratic functions. International Journal of Approximate Reasoning **31**(3) (2002) 313–340
8. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: Bayesian optimization algorithm, population sizing, and time to converge. In: Proceedings of GECCO 2000. (2000) 275–282
9. Pelikan, M.: Bayesian optimization algorithm: From single level to hierarchy. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL (2002) Also IlliGAL Report No. 2002023.
10. Mühlenbein, H., Mahnig, T.: Evolutionary computation and beyond. In Uesaka, Y., et. al. eds.: Foundations of Real-World Intelligence. CLSI Publications (2001) 123–186
11. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press (1995)