Preprint
UCRL-JC-137172

Bayesian Optimization Algorithm, Population Sizing, and Time to Convergence

M. Pelikan, D. E. Goldberg and E. Cantu-Paz

This article was submitted to
Genetic and Evolutionary Computation Conference, Las Vegas, NV,
July 8-12, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

January 19, 2000

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
http://apollo.osti.gov/bridge/

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
http://www.ntis.gov/

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
http://www.llnl.gov/tid/Library.html

Bayesian Optimization Algorithm, Population Sizing, and Time to Convergence

Martin Pelikan, David. E. Goldberg, and Erick Cantú-Paz*
Illinois Genetic Algorithms Laboratory
104 S. Mathews Avenue, Urbana, IL 61801
University of Illinois at Urbana-Champaign
Phone/FAX: (217) 333-2346, (217) 244-5705
`{pelikan,deg,cantupaz}@illigal.ge.uiuc.edu`

Abstract

This paper analyzes convergence properties of the Bayesian optimization algorithm (BOA). It settles the BOA into the framework of problem decomposition used frequently in order to model and understand the behavior of simple genetic algorithms. The growth of the population size and the number of generations until convergence with respect to the size of a problem is theoretically analyzed. The theoretical results are supported by a number of experiments.

1 Introduction

Recently, the Bayesian optimization algorithm (BOA) has proven to optimize problems of bounded difficulty quickly, reliably, and accurately. The number of function evaluations until convergence was investigated on a number of problems. However, the questions of (1) how to choose an adequate population size in order to solve a given problem with a certain degree of accuracy and reliability, and (2) how many generations it will take until the algorithm converges, remained unanswered.

This paper makes an important step toward understanding the mechanics of the BOA. It approximates the growth of the required population size and the number of generations until convergence with respect to the size of a problem by using recent results of the genetic algorithms population-sizing and convergence theory. The theoretical results are supported by a number of experiments.

The paper starts with a brief introduction to the probabilistic model-building genetic algorithms—the evolutionary methods based on building and using probabilistic models of promising solutions in order to guide the further search. Additionally, the motivation and most important results of the population sizing and convergence theory of genetic algorithms are presented. The relationship between the population size and the size of a problem is analyzed in Section 3. The number of generations with respect to the problem size is discussed in Section 4. The paper is summarized and concluded in Section 5.

---

*Also with Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551.

2 Background

This section provides background on the algorithms based on probabilistic modeling of promising solutions and on recent theoretical results of population sizing and time to convergence of simple genetic algorithms.

2.1 Probabilistic Model-Building Genetic Algorithms (PMBGAs)

Probabilistic model-building genetic algorithms (PMBGAs), also called estimation of distribution algorithms (Mühlenbein & Paaß, 1996), replace the genetic recombination of genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989) by building an explicit model of promising solutions and using the constructed model to guide the further search. As models, probability distributions are used. For an overview of recent work on PMBGAs, see Pelikan, Goldberg, and Lobo (1999).

The Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1998) uses Bayesian networks to model promising solutions and subsequently guide the further search. In the BOA, the first population of strings is generated at random. From the current population, the better strings are selected. Any selection method can be used. A Bayesian network that fits the selected set of strings is constructed. Any metric as a measure of quality of networks and any search algorithm can be used to search over the networks in order to maximize/minimize the value of the used metric. Besides the set of good solutions, prior information about the problem can be used in order to enhance the estimation and subsequently improve convergence. New strings are generated according to the joint distribution encoded by the constructed network. The new strings are added into the old population, replacing some of the old ones. The pseudo-code of the BOA can be found in Figure 1.

The Bayesian Optimization Algorithm (BOA)

```
(1) set $t \leftarrow 0$
    randomly generate initial population $P(0)$
(2) select a set of promising strings $S(t)$ from $P(t)$
(3) construct the network $B$ using a chosen metric and constraints
(4) generate a set of new strings $O(t)$ according to the joint distribution encoded by $B$
(5) create a new population $P(t+1)$ by replacing some strings from $P(t)$ with $O(t)$
    set $t \leftarrow t+1$
(6) if the termination criteria are not met, go to (2)
```

Figure 1: The pseudo-code of the Bayesian optimization algorithm.

A Bayesian network is a directed acyclic graph with the nodes corresponding to the variables in the modeled data set (in our case, to the positions in the solution strings). Mathematically, a Bayesian network encodes a joint probability distribution given by

$$p(X) = \prod_{i=0}^{n-1} p(X_i|\Pi_{X_i}), \tag{1}$$

where $X = (X_0, \ldots, X_{n-1})$ is a vector of all the variables in the problem, Π_{X_i} is the set of parents of X_i in the network (the set of nodes from which there exists an edge to X_i) and $p(X_i|\Pi_{X_i})$ is the conditional probability of X_i conditioned on the variables Π_{X_i}. A directed edge relates the variables so that in the encoded distribution, the variable corresponding to the terminal node will be conditioned on the variable corresponding to the initial node. More incoming edges into a node result in a conditional probability of the corresponding variable with conjunctional condition containing all its parents.

Various methods can be used to construct the network given the set of selected solutions. All of these methods have two basic components: a scoring metric which discriminates the networks according to their quality and the search algorithm which searches over the networks to find the one with the best scoring metric value. The BOA can use any scoring metric and any search algorithm. In our recent experiments, we have used the Bayesian-Dirichlet metric (Heckerman, Geiger, & Chickering, 1994). The complexity of the considered models was bounded by the maximum number of incoming edges into any node denoted by k. To search the space of networks, a simple greedy algorithm was used due to its efficiency. For further details, see Pelikan, Goldberg, and Cantú-Paz (1999).

2.2 Decomposing the Problem of Modeling and Understanding GAs

In order to understand and model GAs, the undoubtly complex behavior of GAs has been decomposed into more tractable sub-problems (Goldberg, Deb, & Clark, 1992):

1. Know what the GA is processing: building blocks (BBs).
2. Solve problems tractable by BBs.
3. Supply enough BBs in the initial population.
4. Ensure the growth of necessary BBs.
5. Mix the BBs properly.
6. Decide well among competing BBs.

Even though the BOA uses an explicit model of promising solutions instead of an implicit model brought about by genetic recombination operators in GAs, the goal and mechanics of both algorithms are the same. In fact, the BOA tries to resolve the problems that the GAs have a difficulty of dealing with. Our previous work on the BOA addressed two of the above-listed points that are difficult for conventional GAs: it resulted in the algorithm that ensures a proper growth and mixing of building blocks on problems tractable by BBs. In this paper we address two additional points from the above list: an initial supply of BBs and good decision-making among competing BBs. Our study results in the relationship between an adequate population size and the size of a problem. Additionally, we apply the results of recent work on the time to convergence to the BOA and argue that these theoretical results can be applied directly with little or no modification.

2.3 Population Sizing

Previous estimates of adequate population sizes can be classified into two major categories: (1) models concerned with an initial supply of building blocks (BBs) and (2) models that involve decision-making between competing BBs (Goldberg et al., 1992; Harik et al., 1999).

2.3.1 Having an Adequate Initial Supply of BBs

In order to guarantee a proper reproduction and juxtaposition of BBs, both the GAs and the BOA must have an adequate *initial supply* of BBs. The more good BBs there are in the initial population, the higher the chance of getting the global solution. Assuming that the initial population of solutions is generated with uniform distribution and the solutions are represented by binary strings, the expected number of a particular BB of order k in the initial population is given by

$$m(BB_k) = \frac{n}{2^k} \cdot \tag{2}$$

The first supply model simply considers this estimate and suggests that the population size grows at least exponentially with the size of BBs.

Another study relates the population size and the expected performance of the GA by estimating a number of schemata contained in the population. Holland (1975) estimates the number of schemata in a randomly generated population of N solutions by $O(N^3)$. This estimate was later rederived by Goldberg (1989) by computing the number of schemata in one string and multiplying this by the population size.

2.3.2 Deciding Well Between Competing BBs

Besides having an adequate initial supply of BBs, another important factor that determines the success of GAs is the one of *deciding well* between competing BBs. Naturally, the better building blocks should get more copies in the offspring than the worse ones. However, as it was soon recognized by De Jong (1975), the decision-making for a particular BB is strongly affected by *noise* from the fitness contributions of the remaining building blocks.

The idea of estimating the population size by focusing on one BB and considering the noise from the fitness contributions of the remaining parts of the string solutions resulted in the first population-sizing model introduced by Goldberg, Deb, and Clark (1992). The proposed model reduced decision-making to the two best BBs in a particular partition. It estimated the required population size so that the best BB wins over the second best BB in the same partition. It is natural to expect the best BB to win over other BBs in the same partition as well. This model was a little pessimistic; it required the best BB to win in the first generation. Assuming that the BBs are separable and uniformly scaled, the resulting population size estimate is then given by

$$N = 2c(\alpha)2^k m' \frac{\sigma_{bb}^2}{d^2}, \tag{3}$$

where $c(\alpha)$ is the square of the ordinate of a unit normal distribution where the probability equals to α; α is the probability of failure; k is the order of the considered BB; m' is one less than the number m of BBs in a string (i.e. $m' = m - 1$); σ_{bb}^2 is the root mean square (RMS) fitness variance of the considered BB; and d is the difference between the fitness contributions of the best and the second best BBs.

Harik, Cantú-Paz, Goldberg, and Miller (1999) weakened the assumptions and used the gambler's ruin model in one dimension to model the behavior of the GA so that the best BB could also win in the subsequent generations. Assuming perfect mixing, the population size sufficient to get a solution of certain quality was reduced to

$$N = -2^{k-1} \ln(\alpha) \frac{\sigma_{bb}\sqrt{\pi m'}}{d}, \tag{4}$$

The empirical results with tightly-encoded deceptive building blocks match the theory very well. Thus, with perfect mixing, the required population size in genetic algorithms grows proportionally to the square root of the number of BBs in a problem.

2.4 Time to Convergence

Once we have an adequate population size to get solutions of a required quality, another important question is how many generations it will take the algorithm to converge. Mühlenbein and Schlierkamp-Voosen (1993) estimated the number of generations until convergence for the one-max (also called bit-counting) problem, assuming an infinite population size and perfect mixing, by

$$t_{conv} = \left(\frac{\pi}{2} - \arcsin\left(2p - 1\right)\right) \frac{\sqrt{n}}{I}, \tag{5}$$

where p is the initial proportion of ones on each position, n is the problem size, and I is the selection intensity. The selection intensity in generation t is given by

$$I(t) = \frac{\overline{f}(t+1) - \overline{f}(t)}{\sigma(t)}, \tag{6}$$

where $\overline{f}(t+1)$ is the average fitness of the population in generation $t+1$; $\overline{f}(t)$ is the average fitness in generation t; and $\sigma(t)$ is the standard deviation of the fitness values in generation t. For many commonly used selection schemes (e.g. tournament, truncation, ranking, and (λ, μ) selection) the selection intensity is a constant (Mühlenbein & Schlierkamp-Voosen, 1993). This holds even for fitness functions affected by an external source of normally-distributed noise (Miller & Goldberg, 1996).

The above result suggests that with a population large enough, the number of generations until convergence is proportional to the square root of a problem size and inversely proportional to the selection intensity. The convergence model was later rederived and extended by Miller and Goldberg (1996) to take into account additional normally distributed noise in the fitness function.

In the following text, the solutions are represented by binary strings of fixed length. However, the described techniques can be extended to strings over any finite alphabet. Each position in a string corresponds to a random variable. A particular solution is the corresponding instance of all variables, and a particular population of solutions is the set of instances of all variables.

3 Population Sizing

In order to find solutions of a required quality, the BOA must both (1) find the model necessary to converge to such solutions and (2) have an adequate population size in order to have a sufficient initial supply of BBs and to decide well among competing BBs. Initial supply and decision-making was discussed in Section 2 and, assuming that the model is accurate enough, this theory can be directly applied to the BOA. In the following section, we will concentrate on the accuracy of the model building for additively decomposable problems in the BOA. Only the problems that contain interacting variables are considered, since for the class of linear problems where the model does not contain any interactions the gambler's ruin population-sizing model can be directly used.

3.1 Predicting the Growth of the Population Size

When the proportions of schemata in the first generation are not accurate enough for a model to be correct and accurately capture the dependencies in a problem, the proportions of these schemata in the following generations will deceive the model-building even more. This is caused by the fact that the information that deceived the model-building will be emphasized by the built model which is used to generate the next population of solutions.

Therefore, we expect that it is important to make good decisions in the first generations. This requirement resembles the first population-sizing model of Goldberg, Deb, and Clark (1992) which required the genetic algorithm to make good decisions already in the beginning. The initial population-sizing model was later refined by weakening the assumptions resulting in the claim that the population size should grow proportionally to the square root of the problem size. This suggests that the population-sizing bound derived below for the BOA may be too pessimistic and the reality could be much better. The empirical results presented later in the paper match our theoretical analysis and thus confirm our intuition. We are currently investigating whether the use of incremental model building alleviates the requirement of deciding correct early in the run.

On input, the network-construction algorithm used in the BOA gets the values of the scoring metric for various networks. By observing the changes of the value of the metric while modifying the network by elementary graph operations, the network is incrementally constructed. The resulting network is thus determined by the values of the metric[1]. Assuming that we use the BD metric without pressure toward simpler models and a proper bound on the complexity of the considered models represented by k, the problem is to distinguish correct relationships from fake ones.

As the population size approaches infinity, the BD metric would get perfect information on how the models differ since the probabilities that are on input to the metric (any conditional probabilities of order at most $k+1$) and the corresponding probabilities with an infinitely large population would be the same. The information would be sufficient to make correct independence assumptions and, assuming that the selection pressure is strong enough to identify the dependencies among the bits within some BB, all pairs of bits within any BB would seem dependent on each other. Assuming a sufficient k, the model would be perfect and the assumptions of the gambler's ruin population-sizing model would be satisfied.

However, the problem of insufficiently accurate information in the selected set of solutions emerges with finite population sizes. With finite populations, the noise from fitness contributions of the remaining BBs affects the accuracy of the frequencies of schemata within a particular BB because the structure of the set of selected solutions is determined by the values of the fitness function.

For very small populations, the effect of the noise is very large and the information is insufficient to winnow dependent and independent variables. As the population enlarges the information becomes more and more accurate. In order to quantify the amount by which we have to enlarge the population as the problem size increases, we will look at how the noise in the population changes. For simplicity, only binary tournament selection with replacement will be considered. The approach can be generalized in a straightforward manner. Once we have this result, we will argue that this should be a reasonable estimate due to the nature of the problem. Our results will be supported by a number of experiments with a deceptive function of order three and a trap function of order five.

Let us look at the schema H that is fixed on some positions within one building block. Furthermore, let us denote the expected fitness of the schema H (the average fitness of the schema in an infinite population) by $\overline{f}(H)$. Assuming that the fitness is composed of m independent (separable) uniformly scaled subfunctions with normally distributed fitness, the variance coming from a number of subfunctions is simply the sum of their individual variances. Assuming that the fixed positions in H are in the same subfunction (building block), the variance of the fitness of H can be lower-bounded by $(m-1)\sigma_{bb}^2$ and upper-bounded by the overall variance of the fitness $m\sigma_{bb}^2$, where σ_{bb}^2 is the variance of one sub-function (building block). Therefore, for m not too small, $m\sigma_{bb}^2$

---

[1] In this study we do not consider stochastic network-construction algorithms.

represents quite accurate approximation of the variance of the fitness of H (which is within at most $1/m$ of its real variance). The fitness of schema H is normally distributed, i.e.

$$f(H) \sim N(\overline{f}(H), m\sigma_{bb}^2).$$

The probability of H winning over any of its competitors H' (the schemata with different fixed alleles in the same partition and the don't-care symbols in the remaining locations) in a binary tournament is given by $p(f(H) > f(H'))$, which can be rewritten (Harik et al., 1999) as

$$p(f(H) - f(H') > 0).$$

The fitness of H' is also normally distributed and its variance can be, analogically as for H, approximated by $m\sigma_{bb}^2$, i.e.

$$f(H') \sim N(\overline{f}(H'), m\sigma_{bb}^2).$$

Since both $f(H)$ and $f(H')$ are normally distributed, their difference is also normally distributed with the mean equal to the difference between the means of the fitness distributions of H and H' and the variance equal to the sum of the variances of the two distributions:

$$f(H) - f(H') \sim N(\overline{f}(H) - \overline{f}(H'), 2m\sigma_{bb}^2).$$

The probability $p(f(H) - f(H') > 0)$ together with the probabilities of H winning over its competitors determine the frequency of the schema H in the population of selected solutions. It is the frequencies of different schemata that are used to determine the score for each network and thus guide the construction of the model. Therefore, for a model to be of certain quality, it is natural to expect that these probabilities be within a certain threshold from its expected value, independently of the problem size. For the probability $p(f(H) - f(H') > 0)$ to be within a certain threshold, the sample mean and variance must be close to their expected values. The mean of $f(H) - f(H')$ after doing N tournaments with replacement is within

$$z(\alpha)\frac{\sqrt{2m}\sigma_{bb}}{\sqrt{N}}$$

from its expected value (Sachs, 1984; Goldberg & Rudnick, 1991), where α is the confidence level and $z(\alpha)$ is the critical value of a one-sided normal test of significance at level α. Thus, to get the mean within some accuracy threshold ϵ_1, the following inequality must hold:

$$z(\alpha)\frac{\sqrt{2m}\sigma_{bb}}{\sqrt{N}} \leq \epsilon_1$$

By solving the above inequality for the population size N, we get

$$N \geq 2m\left(\frac{z(\alpha)\sigma_{bb}}{\epsilon_1}\right)^2 \tag{7}$$

Estimating the variance requires approximately (Sachs, 1984)

$$1 + 0.5\left(\frac{z(\alpha)}{\delta}\right)^2$$

samples, where δ is a relative error of the estimated standard deviation from its expected value normalized by the expected value of the variance (i.e. $\delta = (s - \sigma)/\sigma$). For the variance of $f(H) - f(H')$ to be within ϵ_2 from its expected value, we get $\delta = \epsilon_2/(\sqrt{2m}\sigma_{bb})$. An adequate sample size is thus given by

$$1 + m\left(\frac{z(\alpha)\sigma_{bb}}{\epsilon_2}\right).$$

Therefore, the population size in order for the model to be of required quality grows at least linearly with the number of building blocks in a problem. This boundary takes over both the boundary in order to have an adequate initial supply of the BBs as well as the gambler's ruin model and becomes dominant as the problem size increases. The above functional relationship can be analogically obtained by taking any schema spanning through two different BBs. With such schemata, the m from the above estimates would be reduced to $m' = m - 1$ due to the different bounds of the fitness variance of the schema. Nevertheless, this would not change the functional form of our estimate.

3.2 Empirical Results

Experiments presented in this section are performed on two deceptive functions. A deceptive function of order 3 is composed of separable building blocks of order 3, each with a fitness contribution of

$$f_{3deceptive}(u) = \begin{cases} 0.9 & \text{if } u = 0 \\ 0.8 & \text{if } u = 1 \\ 0 & \text{if } u = 2 \\ 1 & \text{otherwise} \end{cases}, \tag{8}$$

where u is the number of ones in the building block. The overall fitness is computed by summing the contributions of individual BBs. The trap function of order 5 is defined as the sum of contributions of $n/5$ distinct building blocks, each of which is given by

$$f_{trap5}(u) = \begin{cases} 4 - u & \text{if } u < 5 \\ 5 & \text{otherwise} \end{cases}, \tag{9}$$

where u is the number of ones in the building block. Both functions have one global optimum in $111\ldots1$ and a deceptive attractor in $000\ldots0$.

In all experiments, the truncation selection with a threshold $\tau = 50\%$ was used (the best half of the population is selected). Offspring replace the worst half of the population. Each run is terminated after the proportion of some bit on each position reaches 95%.

Figure 2 shows the proportion of correct building blocks on a deceptive function of order three found with various population sizes averaged over 100 independent runs. Four problems of sizes $n_1 = 99$, $n_2 = 198$, $n_3 = 297$, and $n_4 = 396$ bits were tested (the size of the problem must be divisible by 3). The number of BBs in the test problems grows linearly ($m_1 = 33$, $m_2 = 66$, $m_3 = 99$, and $m_4 = 132$). Therefore, as the theory suggests, the difference between the population sizes in order to get a solution of a certain quality should be the same between the two smaller and two larger problems. Similar results were obtained on the trap function of order five (see Figure 3). The results confirm that (1) the population size does not grow faster than linearly and (2) our estimate is not overly pessimistic.

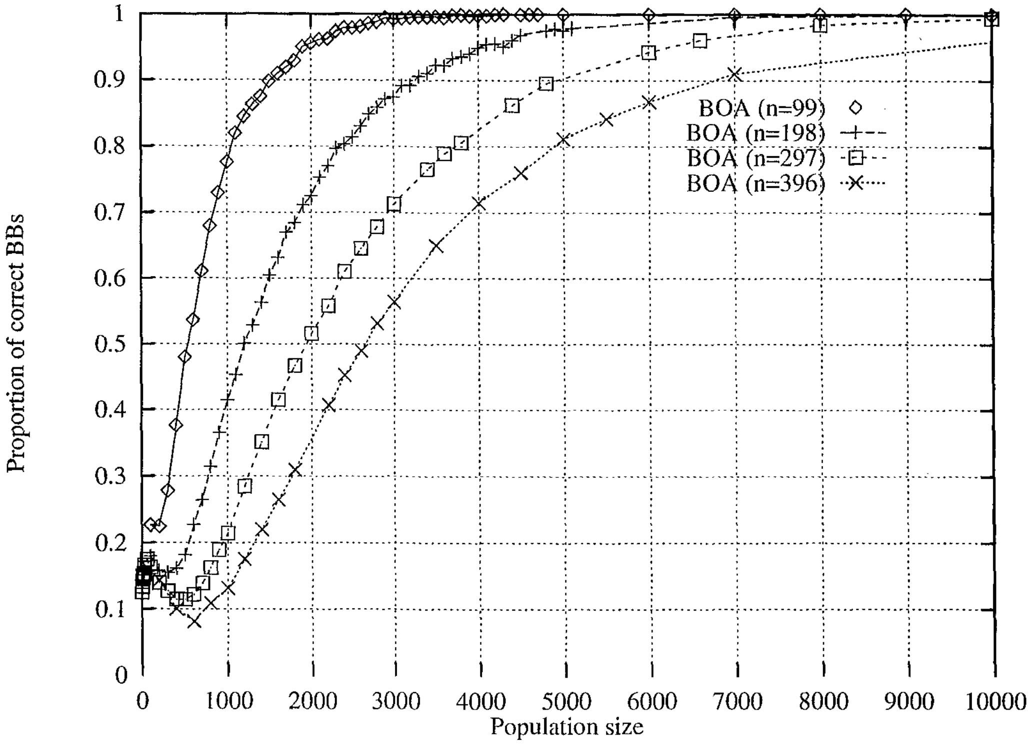

Figure 2: Proportion of correct BBs found by the BOA with the deceptive function of order three of four different sizes.

Interesting behavior can be observed with larger problems. For $n_3 = 297$, for instance, the proportion of correct BBs grows initially for very small populations ($N < 60$), then decreases to its lowest value with population size $N = 500$, and then monotonously increases, decreasing the error as the gambler's ruin model suggests for the simple GA with perfect mixing. In fact, from a certain point, the functional form of the gambler's ruin population sizing model matches the empirical results very accurately. In the next paragraph we will discuss the possible reasons for this rather anomalous behavior, since one would expect that with increasing population size the proportion of correct BBs should increase.

For very small populations, the noise is very high and the frequencies, as a result of highly distorted fitness values, are very inaccurate. That is why in each generation the model is more or less random and the results are similar to what we would get by randomly generating solutions until the population converges due to the elitism. As the difference between expected and real schemata frequencies are reduced, the model becomes more and more stable. However, the information is still insufficiently accurate in order for the model to be correct and the algorithm is deceived to the local optima, leaving only a small proportion of the BBs correct. Even some of the BBs that are correct in the initial population are disrupted and the algorithm is deceived. This behavior can be observed in the algorithms that assume that the variables are independent (e.g. simple GA with uniform crossover and univariate marginal distribution algorithm) whose final solution gets worse with increasing population size. Finally, the model gets more and more accurate, and the proportion of correct BBs grows monotonously with the population size.

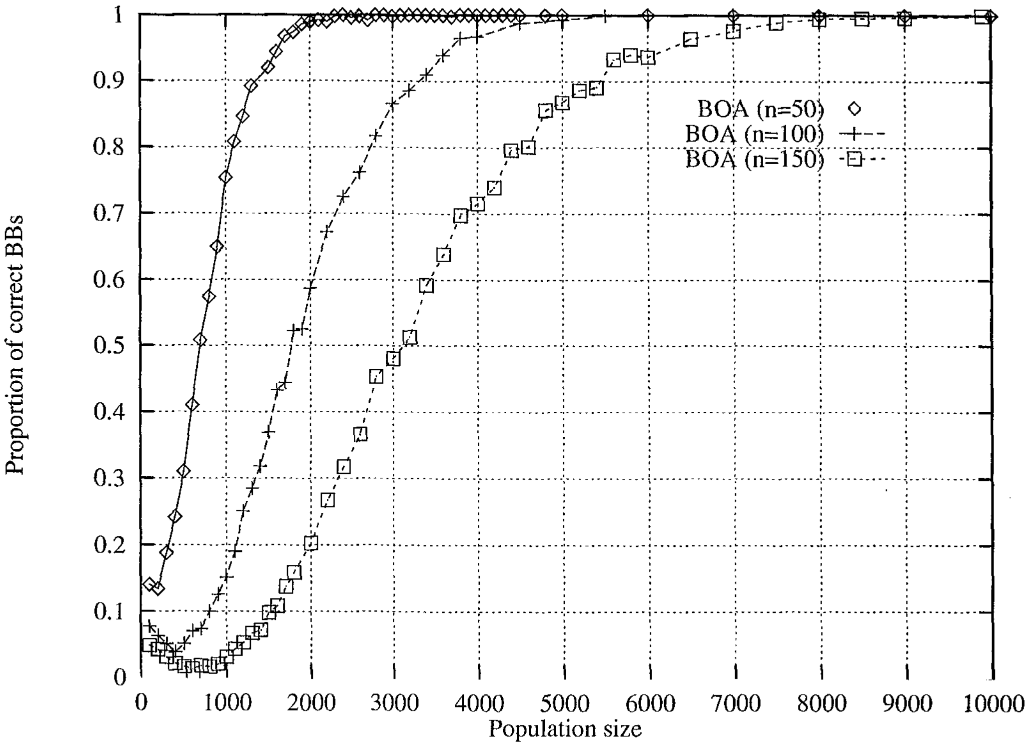

Figure 3: Proportion of correct BBs found by the BOA with the trap function of order five of three different sizes.

4 Number of Generations until Convergence

To estimate the number of generations until convergence, we will assume that (1) the population is large enough for the model to accurately represent all the dependencies and independence relations and (2) a selection scheme with a constant selection intensity (e.g. truncation selection, tournament selection, ranking selection, etc.) is used. Under these assumptions, the convergence model derived by Mühlenbein and Schlierkamp-Voosen (1993) can be used for the onemax as well as for deceptive problems as it was described in Miller and Goldberg (1996), since in both classes of problems the relationship between the variance and bit-wise frequencies is very similar. Thus, even though the size of building blocks varies, the overall behavior can be quite accurately modeled by a bit-wise convergence model with the initial proportion of each bit to $p = 0.5$.

To compare the prediction with the actual empirical values, the two deceptive functions described above were used along with one additional fitness function. The onemax function simply counts the number of ones in the input string.

Figure 4 shows the number of generations until the population has converged and the prediction according to Equation 5 for the three problems. In the experiments, the population is said to have converged when the proportion of some value on each position converges to a single value. Very large populations are used in order to simulate the behavior of the BOA with infinite populations ($N_{low} = 100$ to $N_{high} = 60,000$). Truncation selection with $\tau = 50\%$ was used for which the selection intensity is known to be $I = 0.798$ (Mühlenbein & Schlierkamp-Voosen, 1993) and the offspring replaced the entire parent population. The initial population is generated at random and therefore the initial proportion of ones on each position is set to $p_0 = 0.5$. We expect that a similar

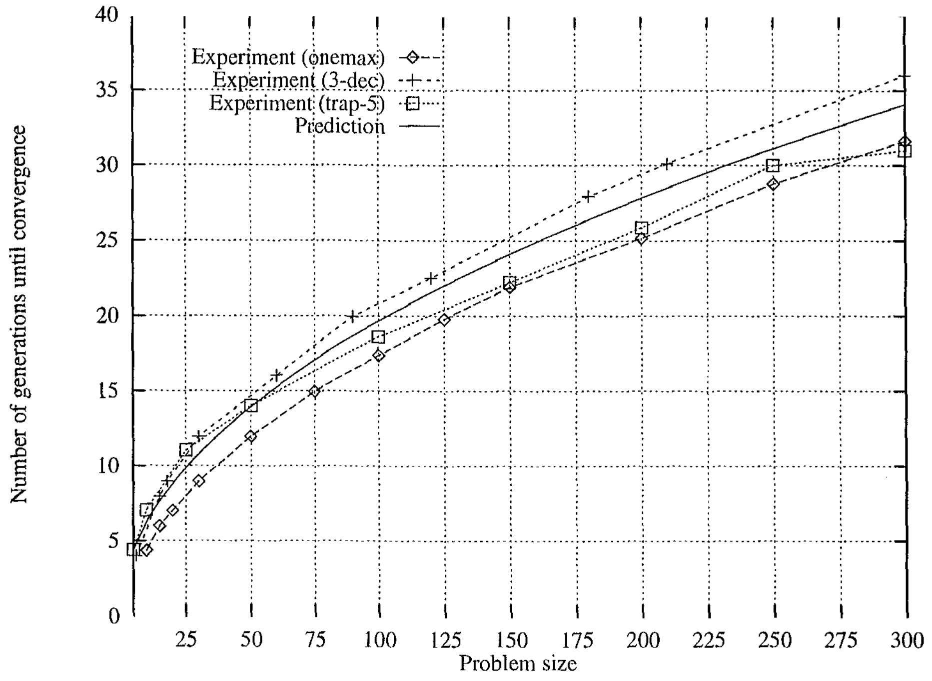

Figure 4: Number of generations until convergence of the BOA with a onemax function, deceptive function of order 3, and a trap function of order 5. The problem sizes range from 0 to 300 bits.

relationship between the problem size and the time to convergence holds for functions whose fitness variance dynamics differs from the one of the simple one-max function.

5 Summary and Conclusions

The paper makes an important step toward understanding the mechanics of the BOA. It approximates the growth of the population size and the number of generations until convergence with respect to the size of a problem. The results suggest that in the BOA the population size grows linearly with the problem size and that the number of generations grows proportionally to the square root of the problem size and inversely proportionally to the selection pressure. The theoretical results are supported by a number of experiments.

The good news is that the overall performance of the BOA for uniformly-scaled separable problems is sub-quadratic. Moreover, the above result should hold for problems with additional noise by simply adding the additional noise to the noise from the fitness function. The bad news is that the population size is larger than the one required by the genetic algorithm with perfect mixing due to increased requirements of the model-building part of the algorithm.

We are currently investigating how exponential scaling of building blocks affects the convergence. We expect that the approximation of an adequate population size should not increase significantly and that the number of generations should be proportional to the problem size like in the simpler algorithms that do not cover mutual interactions of the variables in a problem on the binary integer problem. The genetic drift and the selection pressure are likely be the major factors determining a necessary population size.

6 Acknowledgments

The authors would like to thank Brad L. Miller for useful discussions about convergence models.

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-97-1-0050. Research funding for this work was also provided by the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, the U. S. Army, or the U. S. Government.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48.

References

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Doctoral dissertation, University of Michigan, Ann Arbor. (University Microfilms No. 76-9381).

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, *6*, 333–362.

Goldberg, D. E., & Rudnick, M. (1991). Genetic algorithms and the variance of fitness. *Complex Systems*, *5*(3), 265–278. (Also IlliGAL Report No. 91001).

Harik, G., Cantú-Paz, E., Goldberg, D. E., & Miller, B. L. (1999). The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, *7 (3)*, 231–253.

Heckerman, D., Geiger, D., & Chickering, M. (1994). *Learning Bayesian networks: The combination of knowledge and statistical data* (Technical Report MSR-TR-94-09). Redmond, WA: Microsoft Research.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Miller, B. L., & Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, *4*(2), 113–131.

Mühlenbein, H., & Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. In Eiben, A., Bäck, T., Shoenauer, M., & Schwefel, H. (Eds.), *Parallel Problem Solving from Nature - PPSN IV* (pp. 178–187). Berlin: Springer Verlag.

Mühlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, *1*(1), 25–49.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1998). *Linkage problem, distribution estimation, and Bayesian networks* (IlliGAL Report No. 98013). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith, R. E. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, Volume I (pp. 525–532). Orlando, FL: Morgan Kaufmann Publishers, San Fransisco, CA.

Pelikan, M., Goldberg, D. E., & Lobo, F. (1999). *A survey of optimization by building and using probabilistic models* (IlliGAL Report No. 99018). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Sachs, L. (1984). *Applied statistics* (2nd ed.). Springer Series in Statistics. New York: Springer-Verlag.