Research on the Bayesian Optimization Algorithm

Martin Pelikan and David E. Goldberg

IlliGAL Report No. 2000010
February 2000

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-0897
Fax: (217) 244-5705

Research on the Bayesian Optimization Algorithm

Martin Pelikan and David E. Goldberg
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
{pelikan,deg}@illigal.ge.uiuc.edu

Abstract

This paper summarizes our recent research on the Bayesian optimization algorithm (BOA) and outlines the directions our research in this area has been following. It settles the algorithm in the problem decomposition framework used often to understand the complex behavior of genetic algorithms. It provides the most important research issues to tackle and reviews our recent progress in each of these areas.

1 Introduction

The use of search methods that build models of promising solutions found so far and use the built models to guide the further search has recently shown to result in powerful methods that resolve many of the problems of evolutionary methods identified by the genetic and evolutionary computation community over the past decades.

The purpose of this paper is to review our recent research in this area and outline the background, motivation, and most immediate goals of our research on the Bayesian optimization algorithm. Additionally, the paper settles the BOA into the framework of solving problems tractable by building blocks. Some theoretical and empirical results published elsewhere are provided.

The paper starts by a brief introduction to the use of probabilistic modeling in genetic and evolutionary computation and a brief description of the BOA. In sections 4 and 5, recent results on the growth of the population size and the number of generations with respect to the size of a problem are presented. Section 6 presents motivation to solving problems in a hierarchical fashion and discusses possible directions of future research on this topic. The paper is summarized and concluded in Section 7.

2 Probabilistic Model-Building Genetic Algorithms

Probabilistic model-building genetic algorithms (PMBGAs), also called the estimation of distribution algorithms (Mühlenbein & Paaß, 1996), replace genetic recombination of the 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

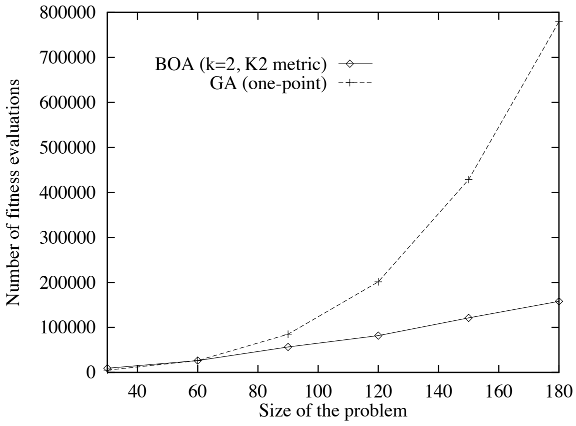


Figure 1: Results for 3-*deceptive* Function.

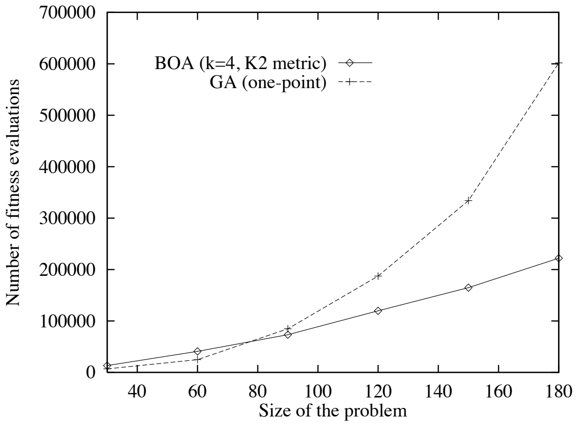


Figure 2: Results for *trap*-5 Function.

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.

As a model of the selected strings, a Bayesian network is used in the BOA. 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

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

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.

To construct the network given the set of selected solutions, various methods can be used. All 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 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).

The BOA has shown to solve problems tractable by BBs quickly, reliably, and accurately. It outperformed the simple genetic algorithms even on problems with tight BBs (the BBs of short defining lengths) which is the "GA-best" case. As the building blocks would get looser, the simple genetic algorithm would perform worse and worse and, eventually, for the BBs spread all through the solution strings, the simple genetic algorithm would require exponential time with respect to the size of a problem. On the other hand, the BOA is independent of the defining length of building blocks and it is capable of identifying the building blocks on the fly by extracting information from the set of promising solutions and using this information to guide its search in order to ensure a proper growth and juxtaposition of the BBs. Some of the comparisons of the BOA and GAs are shown in figures 1 and 2. For both algorithms, the same selection rate and generation gap were used. Other parameters were chosen optimally, based on our empirical experience on each problem. For more details on the experiments, please see Pelikan, Goldberg, and Cantú-Paz (1999).

3 Decomposing the Problem of Modeling and Understanding GAs

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

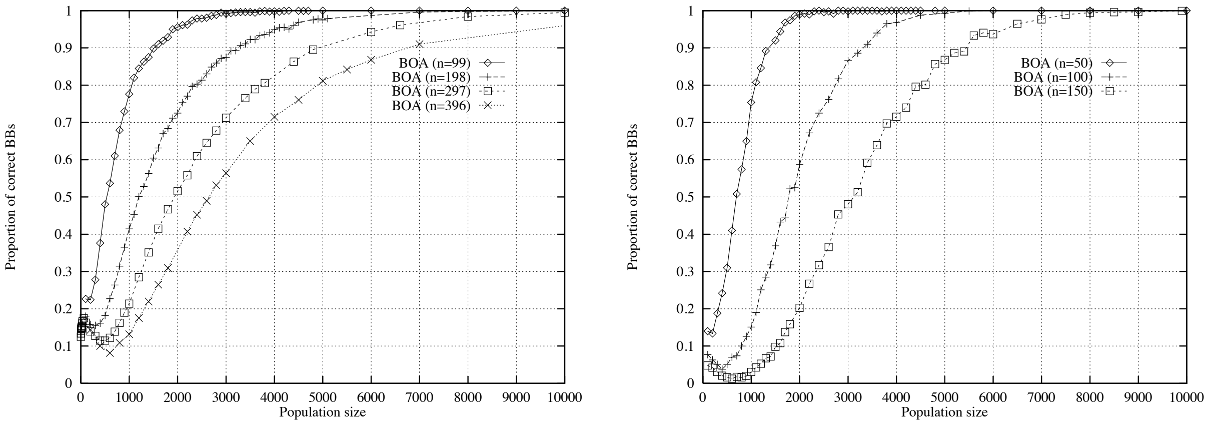
1. Know what the GA is processing: building blocks (BBs).
2. Solve problems tractable by BBs.
3. Supply enough BBs in the initial population.
4. Ensure the growth of necessary BBs.
5. Mix the BBs properly.
6. Decide well among competing BBs.

Without any bias we cannot expect any method to optimize a problem faster than enumeration or random search. To optimize functions efficiently, we must pose some kind of bias to the search. Genetic algorithms bias the search by genetic recombination and mutation to regions that can be reached by combining partial solutions of promising solutions found so far and their close neighborhood. The question is how to combine these solutions effectively and use the information contained in the set of promising solutions best.

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. First, we tackled two of the above-listed points that were difficult to tackle for conventional GAs. Our work resulted in the algorithm that ensures a proper growth and mixing of building blocks on problems tractable by BBs. Later, the initial supply of BBs and the decision-making among competing blocks was discussed and resulted in the relationship between the population size and the size of a problem. By any means, we find the above decomposition a very useful one in order to understand what is going on in the BOA and analyze its behavior more formally.

4 Population Sizing

In order to find the solution of a required quality, the BOA must both (1) find the model necessary to converge to such a solution and (2) have an adequate population size in order to have sufficient initial supply of the BBs and do the decisions among competing BBs well. Initial supply and decision-making was discussed recently (Harik, Cantú-Paz, Goldberg, & Miller, 1999) and, assuming that the model is accurate enough, this theory can be directly applied to the BOA. Our research focused on the accuracy of the model building for additively decomposable problems in the BOA.



(a) Deceptive of order 3

(b) Trap of order 5

Figure 3: Proportion of correct BBs found with various population and problem sizes.

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 do the good decisions in the first generations. This 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. This population-sizing model was later refined by the next population-sizing model that decreased the population estimate by weakening the assumptions and claimed that the population size should grow proportionally to the square root of the problem size. This suggests that our first population-sizing model may have been too pessimistic and the reality would be much better. The empirical results match our theoretical analysis and thus confirm our intuition. We are currently investigating whether the use incremental model building does not alleviate this requirement.

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. Assuming that we use the BD metric without the 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 in order to make correct independence assumptions and, assuming 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

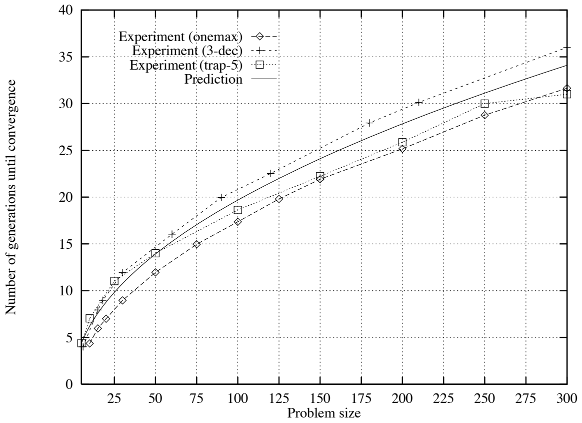


Figure 4: Number of generations until convergence of the BOA with a onemax function, deceptive function of order 3, and a trap function of order 5. The problem sizes range from 0 to 300 bits.

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 fitness function.

In order to quantify the amount by which we have to enlarge the population size for separable problems with uniformly scaled identical subfunctions as the problem size increases, we looked at how the noise in the population was changing. By analyzing the distribution of schemata in the parent population when using a binary tournament selection, the boundary on the population size in order for the frequencies to be within certain error from their expected values was derived. This boundary has two parts, one for the mean and one for the variance of the fitness values in the parent population, which together fully determine the distribution of the parents. Both these boundaries suggest that, to get the solution of certain quality, the population size grows linearly with the problem size. The experiments confirmed our intuition that this is the dominant factor in sizing the populations and that our estimate was not overly pessimistic (see Figure 3).

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. For more information on our work on population sizing in the BOA, please see (Pelikan et al., 2000a).

5 The Number of Generations until Convergence

The number of generations until convergence for the one-max (also called bit-counting) problem, assuming an infinite population size and perfect mixing, is given by(Mühlenbein & Schlierkamp-Voosen, 1993)

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

where p is the initial proportion of ones on each position, n is the problem size, and I is the selection intensity.

To estimate the number of generations until convergence, we assumed that (1) the population was large enough for the model to accurately cover all the dependencies and independence assumptions and (2) a selection scheme with a constant selection intensity (e.g. truncation selection, tournament selection, ranking selection, etc.) was used. Under these assumptions, the convergence model derived by Mühlenbein and Schlierkamp-Voosen (1993) can be used for both onemax as well as 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 $p = 0.5$. Empirical results confirmed the theory and matched our theoretical results very well (see Figure 4).

6 Hierarchical Problem Solving

Recently, the connection between human innovation and genetic algorithms has been discussed (Goldberg, 2000). One of the implications of this argument is that the results of genetic and evolutionary computation can be used as a tool for understanding and modeling human innovation. On the other hand, the achievements and experience from human innovation and engineering design can be seen as yet another source of inspiration for genetic and evolutionary computation in order to design methods that solve hard problems of our interest quickly, accurately, and reliably. One of the latest topics we began to tackle focuses on using *hierarchical problem solving* as one of the cornerstones of engineering design in order to improve current genetic and evolutionary optimization methods.

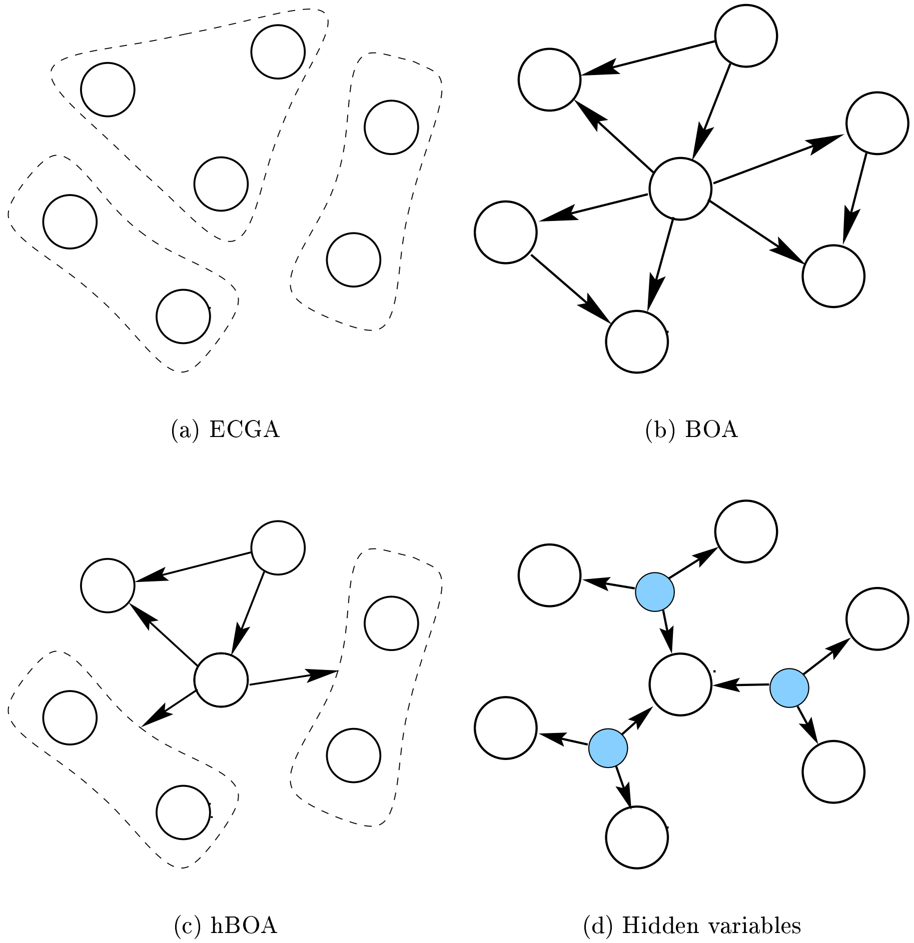
First, the class of the so-called hierarchically decomposable functions, first presented by Goldberg (1998) as the so-called *Tobacco Road Functions*, was extended in order to cover a more general class of problems (Pelikan, Goldberg, & Cantú-Paz, 2000b). This class of problems can be used to test the ability of various algorithms to solve problems in a hierarchical fashion, since it is defined so that the use of hierarchical problem solving pays off there.

In order to adjust modeling to hierarchical problems, one should use models that, among estimating the joint distribution between single variables, also allow multiple variables to be *merged* together and form a *new variable* (Pelikan et al., 2000b). This variable will be further treated as a single unit. In this fashion the solutions of higher order can be formed by using groups (clusters) of variables as basic building blocks.

The idea of clustering the input variables and treating each cluster as an intact building block came from learning used in the extended compact genetic algorithm (ECGA) (Harik, 1999). For each group of variables only instances that are in the modeled data set are considered like in learning Bayesian networks with local structure (Friedman & Goldszmidt, 1999). The clusters (groups) of variables are related as in classical directed-acyclic-graph (DAG) Bayesian networks used in the original BOA algorithm. This class of hybrid models was first introduced by Davies and Moore (1999) who called these models *Huffman networks.*

Let us, for example, at certain point in time, have three positions with only two values in the entire population: 000 and 111. Then, instead of working with each of these positions separately, these can be merged into a single binary variable with two new values $0'$ and $1'$, where $0'$ corresponds to 000 and $1'$ corresponds to 111. In this fashion, both the model complexity as well as the model expressiveness improve. Moreover, by reducing the number of variables, the search for good networks becomes more efficient and accurate. Each group of merged variables represents an intact part of the solutions from lower-level that is to be treated as a single variable on a higher level.

An example model with a few groups of variables is shown in Figure 5c. For comparison, similar examples of models in the BOA and ECGA are shown in parts a) and b) of the same figure. The use of Huffman networks does not require sacrificing modeling generality as in the ECGA. All relationships expressed by DAG models can be covered. On the other side, the overly complex



(a) ECGA

(b) BOA

(c) hBOA

(d) Hidden variables

Figure 5: Models used in a) ECGA, b) BOA, c) hierarchical BOA (Huffman networks), and d) an alternative model based on using hidden variables.

DAG models used in the original BOA can be significantly simplified by "crossing over" the two approaches.

Similar reduction of total model complexity can be achieved by using hidden variables often used in Bayesian networks. In fact, using hidden variables is an alternative and more general approach to the problem of hierarchical model building. We believe that using these models would further improve model-building for problems of a very complex structure. A similar model to the one shown in Figure 5c, based on using hidden variables, is shown in Figure 5d.

Another issue we need to address in order to solve hierarchical problems is to incorporate effective niching in the BOA. Some of the recently proposed niching methods for the simple genetic algorithms can be used. Moreover, the information about the problem structure, encoded by the model of promising solutions, can be used in order to improve the effects of niching. For more information on our work on hierarchical problem solving by the BOA, please see Pelikan et al. (2000b).

7 Summary and Conclusions

This paper summarized our research on the Bayesian optimization algorithm. It presented major results and outlined most important topics of our current research.

One of the important conclusions is that the BOA is a powerful optimization method which can be easily bridged with the existing theory of genetic algorithms and applied to a much wider class of problems. It resolves major problems of the GA application in real world—a proper growth and mixing of building blocks. Moreover, the overall performance still remains sub-quadratic in a number of decision variables in a problem at hand. By approaching problems in a hierarchical fashion, a completely new class of problems can be solved efficiently.

Acknowledgments

The 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.

References

Davies, S., & Moore, A. (1999). Using bayesian networks for lossless compression in data mining. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99)* (pp. 387–391). San Diego, CA: ACM Press.

Friedman, N., & Goldszmidt, M. (1999). Learning Bayesian networks with local structure. In Jordan, M. I. (Ed.), *Graphical Models* (1 ed.). (pp. 421–459). Cambridge, MA: MIT Press.

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

Goldberg, D. E. (1998, June 15). Four keys to understanding building-block difficulty. Presented in Projet FRACTALES Seminar at I.N.R.I.A. Rocquencourt, Le Chesnay, Cedex.

Goldberg, D. E. (2000). The design of innovation: Lessons from genetic algorithms, lessons for the real world. *Technological Forecasting and Social Change*. In press.

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

Harik, G. (1999). *Linkage learning via probabilistic modeling in the ECGA* (IlliGAL Report No. 99010). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

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., & Cantú-Paz, E. (2000a). *Bayesian optimization algorithm, population sizing, and time to convergence* (IlliGAL Report No. 2000001). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000b). *Hierarchical problem solving by the bayesian optimization algorithm* (IlliGAL Report No. 2000002). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

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.