1

Parameter-less Hierarchical Bayesian Optimization Algorithm

Martin Pelikan[1], Alexander Hartmann[2], and Tz-Kai Lin[1]

[1] Dept. of Math. and Computer Science, 320 CCB
University of Missouri at St. Louis
8001 Natural Bridge Rd., St. Louis, MO 63121
pelikan@cs.umsl.edu
tlkq4@studentmail.umsl.edu

[2] Institut für Theoretische Physik
Universität Göttingen
Friedrich-Hund-Platz 1
37077 Göttingen, Germany
hartmann@physik.uni-goettingen.de

Summary. This chapter describes the parameter-less hierarchical Bayesian optimization algorithm (hBOA), which enables the use of hBOA without the need for tuning parameters for solving each problem instance. There are three crucial parameters in hBOA: (1) the selection pressure, (2) the window size for restricted tournaments, and (3) the population size. Although both the selection pressure and the window size influence hBOA performance, the asymptotic time complexity of hBOA should not change with typical choices of these two parameters. However, there is no population size that is suitable for all problems of interest because the population size strongly depends on problem size and problem difficulty. To eliminate the population size, the parameter-less hBOA adopts the population-sizing technique of the parameter-less genetic algorithm. Based on the existing theory, the parameter-less hBOA should be able to solve nearly decomposable and hierarchical problems in a quadratic or subquadratic number of function evaluations without the need for setting any parameters whatsoever. A number of experiments are presented to verify scalability of the parameter-less hBOA. The chapter also discusses how the parameter-less population-sizing scheme can be incorporated into other estimation of distribution algorithms (EDAs).

Key words: Hierarchical BOA, population sizing, parameter-less evolutionary algorithms, spin glass, trap, hierarchical trap.

1.1 Introduction

When the hierarchical Bayesian optimization algorithm (hBOA) was designed [29, 31], it was argued that hBOA can solve difficult nearly decomposable and hierarchical problems without the need for setting any parameters. As a result, to solve a new optimization problem, it should be sufficient to plug the new problem into hBOA, press the start button, and wait until hBOA figures out where the optimum is. It was argued that all hBOA parameters except for the population size can be set to their default values without affecting good scalability of hBOA [36, 27, 36]. However, all experiments assumed that the population size is set optimally to obtain best performance while retaining reliable convergence [36, 27]. That is why the dream of having a fully parameter-less optimizer for the entire class of nearly decomposable and hierarchical problems remained one parameter away from reality.

The purpose of this chapter is to propose a *fully parameter-less hBOA* by implementing the parameter-less population-sizing scheme of the parameter-less genetic algorithm (GA) [11] in hBOA. The parameter-less hBOA simulates a collection of populations of different sizes in parallel, starting with a small, base population size. Each next population is twice as large as the previous one. Each population is allowed to proceed for at least twice as many iterations as the next bigger population. This ensures that although there is no upper bound on the number of populations simulated in parallel and on their size, the overall computational overhead is still reasonable compared to the case with an optimal population size. In fact, theory exists that shows that the parameter-less population sizing scheme in the parameter-less GA does not increase the number of function evaluations until convergence by more than a logarithmic factor [35]. As a result, hBOA can be expected to perform within a logarithmic factor from the case with the optimal population size. This is verified with experiments on nearly decomposable and hierarchical problems. The parameter-less hBOA is also tested in a hybrid algorithm created by combining the parameter-less hBOA with cluster exact approximation (CEA) [15, 16] on the problem of finding ground states of two-dimensional Ising spin glasses with $\pm J$ couplings and periodic boundary conditions. The results indicate that although the performance of the parameter-less hBOA+CEA is expected to be within a logarithmic factor of the performance of hBOA+CEA with optimal population size, the hybrid setting may increase the difference between these two cases by affecting the division of labor between the local and global searchers.

The chapter starts with a brief description of the hierarchical BOA (hBOA) in Section 1.2. Section 1.3 describes the parameter-less hBOA. Section 1.4 describes, presents and discusses experimental results. Section 1.5 outlines how the parameter-less population-sizing scheme can be used in many other estimation of distribution algorithms (EDAs). Finally, Section 1.6 summarizes and concludes the chapter.

1.2 Hierarchical Bayesian Optimization Algorithm

EDAs [24, 32, 18, 4] replace traditional variation operators of genetic and evolutionary algorithms by a two-step procedure:

1. Build a probabilistic model of promising solutions after selection.
2. Sample the built probabilistic model to generate new solutions.

By replacing standard crossover and mutation operators with machine learning techniques that allow automatic discovery of problem regularities from populations of promising solutions, EDAs can solve broad classes of difficult problems quickly, accurately, and reliably, including many problems intractable using other optimization methods [29, 31].

For an overview of EDAs, please see references [32] and [18]. EDAs are also referred to as the probabilistic model-building genetic algorithms (PMBGAs) [32] and iterated density-estimation algorithms (IDEAs) [3]. The remainder of this section describes the hierarchical Bayesian optimization algorithm (hBOA) [29, 31, 27], which is one of the most advanced and powerful EDAs.

1.2.1 Hierarchical BOA (hBOA)

The hierarchical Bayesian optimization algorithm (hBOA) [29, 31, 27] evolves a population of candidate solutions represented by fixed-length discrete strings over a finite alphabet. Usually, binary strings are considered but any other finite-base alphabet can be used analogically.

The first population of candidate solutions is usually generated at random according to the uniform distribution over all candidate solutions. The population is updated for a number of iterations using two basic operators: (1) selection and (2) variation. The selection operator selects better solutions at the expense of the worse ones from the current population, yielding a population of promising candidates. In this chapter, binary tournament selection is used. Binary tournament selection selects one solution at a time by first choosing two random candidate solutions from the current population and then selecting the best solution out of this subset as a member of the new population. Random tournaments are repeated until the new selected population has the same size as the original population and thus each candidate solution is expected to participate in two tournaments.

The variation operator starts by learning a probabilistic model of the selected solutions. hBOA uses Bayesian networks with local structures [5] to model promising solutions. The variation operator then proceeds by sampling the probabilistic model to generate new solutions.

The new solutions are incorporated into the original population using the restricted tournament replacement (RTR) [12], which ensures that useful diversity in the population is maintained for long periods of time. Specifically, RTR incorporates each new solution X using the following three steps:

```
Hierarchical BOA (hBOA)
  t := 0;
  generate initial population P(0);
  while (not done) {
    select population of promising solutions S(t);
    build Bayesian network B(t) with local struct. for S(t);
    sample B(t) to generate offspring O(t);
    incorporate O(t) into P(t) using RTR yielding P(t+1);
    t := t+1;
  };
```

Fig. 1.1. The pseudocode of the hierarchical BOA (hBOA).

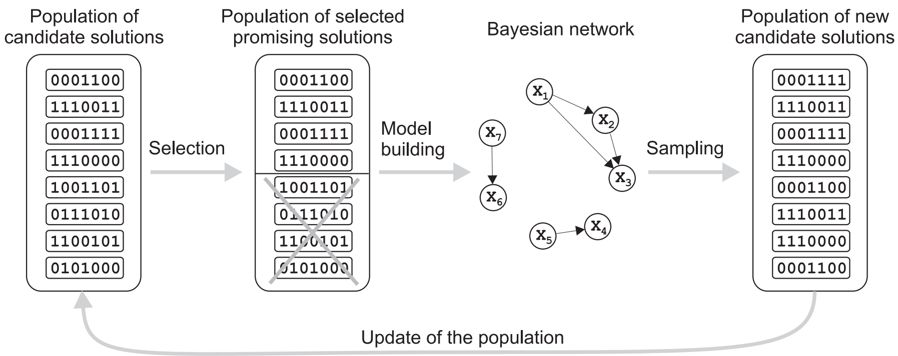

Fig. 1.2. hBOA starts by generating the initial population of candidate solutions at random. In each iteration, hBOA generates new solutions by sampling a Bayesian network built for the selected population of candidate solutions. The iteration concludes by incorporating new solutions into the original population using restricted tournament replacement. The new iteration is executed if the termination criteria are not met.

1. Select a random subset W of w solutions from the original population.
2. Denote Y the candidate solution in W that is most similar to X.
3. X replaces Y is it is better; otherwise, X is discarded.

A robust rule of thumb is to set $w = \min\{n, N/20\}$ where n is the number of decision variables in the problem and N is the population size [30, 27].

The run is terminated when a good enough solution has been found, when the population has not improved for a long time, or when the number of generations has exceeded a given upper bound. Figure 1.1 shows the pseudocode of hBOA. The procedure of hBOA is illustrated in figure 1.2. For a more detailed description of hBOA, see [29, 26, 27].

1.3 Parameter-less hBOA

The parameter-less genetic algorithm (GA) [11] eliminates the need for setting parameters—such as the population size, selection pressure, crossover rate, and mutation rate—in genetic algorithms. The crossover rate and selection pressure are set to ensure consistent growth of building blocks based on the schema theorem. The mutation rate can be eliminated in a similar manner. The population size is eliminated by simulating a collection of populations of different sizes. In the context of hBOA, eliminating the population size is the most important part of the parameter-less GA, because the selection pressure influences performance of hBOA by only a constant factor and there are no crossover or mutation rates in hBOA.

The parameter-less hBOA incorporates the population-sizing technique of the parameter-less GA into hBOA by simulating a collection of populations

$$P = \{P_0, P_1, \ldots\}.$$

The size of the first population P_0 is denoted by N_0 and it is called the base population size. The size of the population P_i is denoted by N_i and it can be obtained by multiplying the base population size by a factor of 2^i:

$$N_i = N_0 2^i.$$

For all $i \in \{1, 2, \ldots\}$, one generation of P_i is executed after executing $k \geq 2$ generations of P_{i-1}. Here we use $k = 2$ so that all populations proceed at the same speed with respect to the number of evaluations. Each population is initialized just before its first iteration is executed. The pseudocode that can be used to simulate the collection of populations described here is shown in Figure 1.3. This implementation is slightly different from the one based on a k-ary counter described in the first parameter-less GA study [11].

1.3.1 When to Terminate a Population?

To terminate populations in the collection, the parameter-less hBOA can use the same termination criteria as the parameter-less GA [11]. However, since hBOA uses niching, no population in the collection can be expected to converge for a long time. To improve performance of the parameter-less hBOA, we upper-bound the number of generations for each population using the estimates predicted by theory [25, 39, 10, 40, 36, 27] so that the number of iterations is at least several times greater than necessary to solve each problem.

Nonetheless, similarly as in the case of the parameter-less GA [35, 34], the parameter-less hBOA can run all populations indefinitely long without increasing the logarithmic growth of the factor by which the computational complexity of hBOA increases compared to the case with the optimal population size.

```
Parameter-less hBOA
  initialize P[0];
  generation[0]=0;
  max_initialized=0;
  i=0;
  while (not done) {
    simulate one generation of P[i];
    generation[i] = generation[i]+1;
    if (generation[i] mod k = 0) {
      i = i + 1;
      if (i>max_initialized) {
        initialize P[i];
        max_initialized=i;
      }
    }
    else
      i = 0;
  };
```

Fig. 1.3. The pseudocode of the parameter-less hBOA.

1.4 Experiments

This section describes experiments, presents experimental results, and discusses the results.

1.4.1 Description of Experiments

Both hBOA and the parameter-less hBOA have been applied to artificial hierarchical and nearly decomposable problems and 2D $\pm J$ spin glasses with nearest neighbor interactions and periodic boundary conditions. Standard hBOA uses the minimum population size required for reliable convergence to the global optimum, whereas the parameter-less hBOA uses the parameter-less population sizing scheme with the base population size $N_0 = 10$.

For all problems, problem size was varied to investigate the scalability of both algorithms. For spin glasses, systems of different size up to 2500 spins were tested and 1000 random instances were examined for each problem size to ensure that the results would provide an insight into hBOA performance on a wide range of spin glass instances. To enhance efficiency of hBOA on spin glasses, cluster-exact approximation is used to improve each candidate solution in the population [15, 33].

For each problem instance and problem size, the parameter-less hBOA is first run to find the optimum in 100 independent runs and the total number of evaluations in every run is recorded. The average number of function evaluations is then displayed. These results are compared to the results for hBOA with the minimum population size that ensures that 30 independent runs converge to the optimum. The minimum population size is determined using bisection until the width of the resulting interval is at most 10% of the lower bound [26, 27].

In all experiments, binary tournament selection without replacement is used to select promising solutions, and RTR is used as a replacement strategy where the window size is $w = \min\{n, N/20\}$.

The remainder of this section discusses test problems and experimental results.

1.4.2 Concatenated Trap of Order 5 (trap-5)

In concatenated traps of order 5 (trap-5) [1, 8], the number of bits in solution strings must be divisible by 5. The input string is first partitioned into independent groups of 5 bits each. This partitioning is unknown to the algorithm, but it does not change during the run. A 5-bit trap function is applied to each group of 5 bits and the contributions of all traps are added together to form the fitness.

The 5-bit trap used to compute the contribution of each group of 5 bits is defined as

$$trap_5(u) = \begin{cases} 5 & \text{if } u = 5 \\ 4 - u & \text{if } u < 5 \end{cases}, \tag{1.1}$$

where u is the number of 1s in the input string of 5 bits. Trap-5 has one global optimum in the string of all ones and $(2^{n/5} - 1)$ other local optima. Trap-5 is fully deceptive [7]. Consequently, variation operators should not break interactions between the bits in each group because all statistics of lower order lead away from the optimum.

Because trap-5 is a fully deceptive decomposable problem of bounded difficulty, this function provides a good test for optimization algorithms that address decomposable problems of bounded difficulty. If an algorithm is not capable of identifying appropriate problem decomposition, it should be mislead away from the optimum. That is why standard crossover and mutation operators fail to solve trap-5 scalably [10, 38, 23].

Figure 1.4 shows the number of evaluations of hBOA with the optimal population size determined by the bisection method and the parameter-less hBOA on trap-5 of $n = 25$ to $n = 150$ bits. The results indicate that the number of function evaluations for decomposable problems increases only by a nearly constant factor, and eliminating the parameters of hBOA thus does not seem to affect hBOA scalability on decomposable problems qualitatively as would be expected from the theory of the parameter-less population sizing scheme [35].

1.4.3 Hierarchical Traps of Order 3 (hTrap)

In an order-3 hierarchical trap (hTrap) with L levels [29, 28], the total number of bits in a candidate solution is $n = 3^L$. A candidate solution is evaluated on

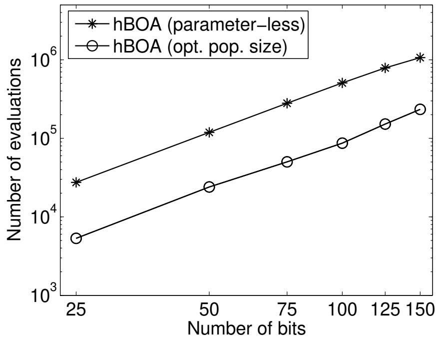

Fig. 1.4. Parameter-less hBOA and hBOA with the optimal population size on order-5 traps of $n = 25$ to $n = 150$ bits.

multiple levels and the overall value of the hierarchical trap is computed as the sum of contributions on all levels.

In the variant of hTrap used in this work, on the lowest level, groups of 3 bits contribute to the overall fitness using a generalized 3-bit trap

$$trap_3(u) = \begin{cases} f_{high} & \text{if } u = 3 \\ f_{low} - u\frac{f_{low}}{2} & \text{otherwise} \end{cases}, \tag{1.2}$$

where $f_{high} = f_{low} = 1$.

Each group of 3 bits corresponding to one of the traps is then mapped to a single symbol on the next level; a 000 is mapped to a 0, a 111 is mapped to a 1, and everything else is mapped to the null symbol '-'. The bits on the next level again contribute to the overall fitness using 3-bit traps defined above, and the groups are mapped to an even higher level. This continues until the top level is evaluated that contains 3 bits total. However, on the top level, a trap with $f_{high} = 1$ and $f_{low} = 0.9$ is applied. Any group of bits containing the null symbol does not contribute to the overall fitness.

To make the overall contribution at each level of the same magnitude, the contributions of traps on ith level from the bottom are multiplied by 3^i.

hTraps have many local optima, but only one global optimum in the string of all ones. Nonetheless, any single-level decomposition into subproblems of bounded order will lead away from the global optimum. That is why hTraps necessitate an optimizer that can build solutions hierarchically by juxtaposing good partial solutions over multiple levels of difficulty until the global optimum if found.

Figure 1.5 shows the number of evaluations of the hierarchical BOA with optimal population size determined by the bisection method and the parameter-less hBOA on hierarchical traps of $n = 27, 81, 243,$ and 729 bits. The results indicate that the number of function evaluations for hierarchically

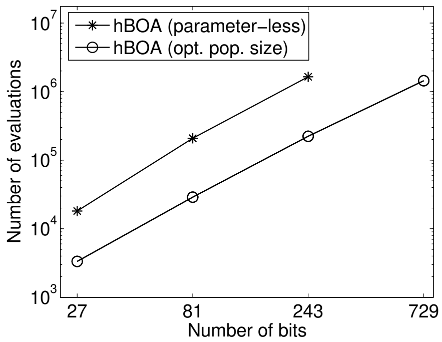

Fig. 1.5. Parameter-less hBOA and hBOA with the optimal population size on the hierarchical trap of $n = 27$ to $n = 729$ bits.

decomposable problems increases only by a constant factor, and eliminating the parameters of hBOA does not affect scalability of hBOA on this class of problems qualitatively.

1.4.4 Ising Spin Glasses

Ising spin glasses provide a class of prototypical models for disordered systems and have played a central role in statistical physics during the last three decades [2, 21, 9, 41]. Here we discuss a simple two-dimensional Ising spin glass model, which provides a rich class of difficult optimization problems that are intractable with most standard crossover and mutation operators [27] as well as state-of-the-art Markov chain Monte Carlo methods [6]. There are three primary sources of problem difficulty of Ising spin glasses: (1) Ising spin glasses contain a large number of local optima, which often grows exponentially with problem size, (2) local optima in the objective-function landscape are surrounded by low-quality points, and (3) effective exploration of the space of candidate solutions (spin configurations) necessitates that the exploration operator considers interactions between a large number of spins (problem variables).

A 2D spin-glass system consists of a regular 2D grid containing n nodes where each node i corresponds to a spin s_i and each edge $\langle i, j\rangle$ corresponds to a coupling between two spins s_i and s_j. Each edge has a real value associated with it that defines the relationship between the two connected spins. To approximate the behavior of the large-scale system, periodic boundary conditions are often used that introduce a coupling between the first and the last element along each dimension.

For the classical Ising model, each spin s_i can be in one of two states: $s_i = +1$ or $s_i = -1$. Note that this simplification corresponds to highly anisotropic systems, which do indeed exist in some experimental situations.

Nevertheless, the two-state Ising model comprises all basic effects also found in models with more degrees of freedom. A specific set of coupling constants define a spin glass instance. Each possible setting of the states of all spins is called a spin configuration.

Given a set of coupling constants $J_{i,j}$, and a configuration of spins $C = \{s_i\}$, the energy can be computed as

$$E(C) = \sum_{\langle i,j \rangle} s_i J_{i,j} s_j \ , \tag{1.3}$$

where $i, j \in \{0, 1, \ldots, n-1\}$ and $\langle i,j \rangle$ denote the nearest neighbors in the underlying grid (allowed edges). Couplings with $J_{i,j} > 0$ favour neighbouring spins to have opposite direction $s_i = -s_j$, i.e. this yields a negative energy contribution $s_j J_{ij,} s_j < 0$. In this case the coupling is called *satisfied*. Couplings $J_{i,j} < 0$ are satisfied if $s_i = s_j$. For most possible sets of coupling constants, so called *frustration* occurs. This means that it is impossible to find a spin configuration that satisfies *all* couplings at the same time.

Here we consider the task of finding *ground states* [17] of a 2D spin glass with specified coupling constants, where a ground state is the configuration of spins that minimizes the energy of the system given by equation 1.3, i.e. it minimizes the weighted sum of the unsatisfied couplings.

Each spin glass configuration is represented by a string of bits where each bit corresponds to one spin: a 0 represents a spin -1, and a 1 represents a spin $+1$. We created 1000 random spin glasses of sizes 20×20 (400 spins) to 50×50 (2500 spins) by generating coupling constants to be $+1$ or -1 with equal probabilities. Spin glasses with coupling constants restricted to $+1$ and -1 are called $\pm J$ spin glasses. The optimum for most 2D instances was verified with the branch-and-cut algorithm provided at the Spin Glass Ground State Server at the University of Köln [37].

To enhance hBOA efficiency, cluster exact approximation (CEA) [15, 16] is incorporated into hBOA to improve all solutions in the population similarly as in [33]. CEA proceeds by selecting randomly a maximal cluster of spins such that the subset of couplings between these spins is non-frustrated. Then CEA changes the cluster spins optimally given the fixed values of all remaining spins. This can be done efficiently using the maximum-flow algorithm [17]. CEA can be repeated a number of times because the clusters of selected spins tend to differ and so do the effects of the corresponding updates.

In the hybrid algorithm hBOA+CEA created by combining hBOA and CEA, before evaluating each spin glass configuration in hBOA, CEA is applied to improve the configuration for a number of steps. The application of CEA to each configuration is terminated once the configuration cannot be improved for a certain number of iterations; more specifically, the bound on the number of failures is $\sqrt{n}$ where n is the number of spins. Since one iteration of CEA can be expected to be at least as expensive as one evaluation of a configuration [22], instead of counting the number of evaluated candidate solutions, we consider

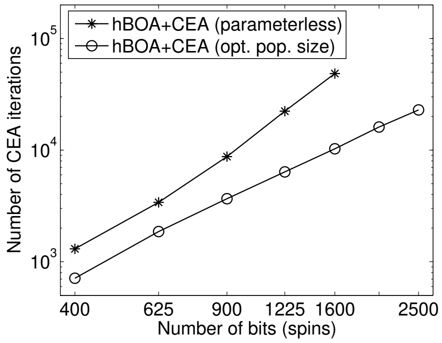

Fig. 1.6. Parameter-less hBOA and hBOA with the optimal population size on 2D Ising spin glasses with random $\pm J$ couplings and periodic boundary conditions. Both methods are combined with cluster exact approximation (CEA).

the number of CEA iterations. The overall number of CEA iterations ran on candidate solutions during the entire run represents the time complexity of this run.

Figure 1.5 shows the number of CEA iterations for hBOA+CEA with the optimal population size determined by the bisection method and the parameter-less hBOA+CEA on 2D $\pm J$ spin glasses. The good news is that the number of CEA iterations for the parameter-less hBOA still grows as a low-order polynomial with respect to the number of decision variables. The bad news is that, unlike for single-level and hierarchical traps, in this case the parameter-less population sizing does influence the asymptotic complexity of hBOA. Although the number of CEA iterations seems to be still upper-bounded by a low-order polynomial, the order of the polynomial that approximates the number of evaluations increases slightly.

At the first sight, the increase in the order of the polynomial that approximates the asymptotic growth of the number of CEA steps seems to disagree with the existing theory, which claims that the factor should be at most logarithmic [35]. One way to explain these results is to realize that for hybrid variants of hBOA, the time complexity of evaluating a candidate solution depends on the solution itself. For any population, early in the run, configurations are random or nearly random, Hence, it can be expected to be rather easy to improve most configurations with CEA and each solution will most likely lead to a high number of CEA iterations until no further improvements result in a termination of the CEA iteration. Later in the run most configurations should be near a local optimum and CEA would usually lead to no improvement or only little improvement at a time, hence the CEA iteration terminates quickly. Since in the parameter-less hBOA many random and nearly random solutions are evaluated in a number of populations, the number of CEA iterations per evaluation of a candidate solution is expected to increase, leading to the increase of the asymptotic complexity of hBOA+CEA using the parameter-less population sizing. Similar behavior can be observed for hBOA hybrids with simpler local search methods although the reasons for that behavior are slightly different [34].

1.5 Eliminating Population Size in Other EDAs

Eliminating the parameters of hBOA is important on its own right, but an important question is whether the proposed scheme can be used to eliminate parameters of other EDAs. Intuitively, the same approach can be used to eliminate the need for setting the population size in most other EDAs. We present two examples that should apply to many EDAs.

1.5.1 Eliminating Population Size in Population-Based EDAs

Any EDA that maintains a population of candidate solutions can be modified by incorporating the parameter-less population-sizing scheme of the parameter-less genetic algorithm analogically to the parameter-less hBOA. For example, the parameter-less population-sizing can be used in the extended compact genetic algorithm (ECGA) [19]. Nonetheless, if there are other parameters that crucially depend on the problem, such as the standard deviation of Gaussian kernel distributions, more parameters would have to be eliminated to create a true parameter-less EDA.

1.5.2 Parameter-less Compact Genetic Algorithm

Although some EDAs do not maintain an explicit population of candidate solutions, they represent a population implicitly using a probabilistic model; this probabilistic model is updated based on the results of tournaments between candidate solutions generated according to the model. Despite the absence of the population, there is usually a parameter that can be seen as a counterpart of the population size and it is just as important as the population size in population-based EDAs. One of the algorithms that use incremental updates of the probabilistic model is the compact genetic algorithm [13, 14].

In the compact genetic algorithm (cGA) [13, 14], the population is replaced by a probability vector that stores the estimated probability of a 1 in each string position. The probability-vector entries are initialized to 0.5 (uniform distribution). In each iteration, the probability-vector entries are updated by performing a binary tournament between two solutions generated according to the probability vector and modifying vector entries for the bits that differ in the two competing solutions. The update rule depends on the learning rate, which corresponds to the effective population size.

The parameter-less cGA can be created by replacing the multiple populations of the parameter-less genetic algorithm by probability vectors; each

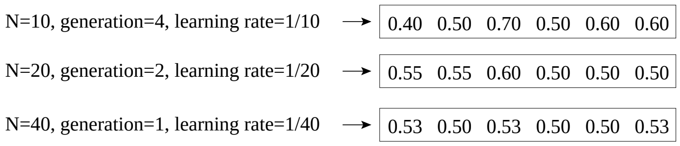

Fig. 1.7. Probability vectors of the parameter-less compact genetic algorithm after running the first iteration on the third probability vector for a 6-bit onemax. Since the optimum of onemax is in the string of all ones, it is expected that the proportions of ones increase over time.

population is represented by one probability vector [20]. In each probability vector, the learning rate $1/N$ is used to represent the population size N. Using the population sizes that grow with an integer power of two, this leads to a series of exponentially decreasing learning rates. See figure 1.7 for an illustration of this concept.

All probability vectors can be simulated indefinitely, similarly as in the parameter-less hBOA. Another approach is to terminate each probability vector when its fitness average over a specific period of time becomes lower than the fitness average for the probability vector representing a larger population size.

Since the learning rate is the only parameter of cGA, the parameter-less cGA is a true parameter-less algorithm. A similar approach could be used to update the learning rate in other similar EDAs, such as PBIL, although the update of the appropriate parameters may not be as straightforward as in cGA.

1.6 Summary and Conclusions

This chapter described, implemented and tested the parameter-less hierarchical Bayesian optimization algorithm (hBOA). The parameter-less hBOA enables the practitioner to simply plug the problem into hBOA without requiring the practitioner to first estimate an adequate population size or other problem-specific parameters. Despite the parameter-less scheme, low-order polynomial time complexity of hBOA on broad classes of optimization problems is retained. The parameter-less hBOA is thus a true black-box optimization algorithm, which can be applied to hierarchical and nearly decomposable problems without setting any parameters whatsoever.

The parameter-less population-sizing scheme can be used in other estimation of distribution algorithms, including the algorithms that replace the population by a probabilistic model and use incremental updates of the probabilistic model.

Acknowledgments

This work was supported by the Research Award and the Research Board at the University of Missouri, and by the *VolkswagenStiftung* (Germany) within the program "Nachwuchsgruppen an Universitäten". Most experiments were done using the hBOA software developed by Martin Pelikan and David E. Goldberg at the University of Illinois at Urbana-Champaign. The experiments presented in this paper were completed at the Beowulf cluster at the University of Missouri at St. Louis.

References

[1] Ackley, D. H. (1987). An empirical study of bit vector function optimization. *Genetic Algorithms and Simulated Annealing*, pages 170–204.

[2] Binder, K. and Young, A. (1986). Spin-glasses: Experimental facts, theoretical concepts and open questions. *Rev. Mod. Phys.*, 58:801.

[3] Bosman, P. A. N. and Thierens, D. (2000). Continuous iterated density estimation evolutionary algorithms within the IDEA framework. *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 197–200.

[4] Cantú-Paz, E., Pelikan, M., and Sastry, K., editors (2006). *Scalable optimization via probabilistic modeling: From algorithms to applications.* Springer-Verlag.

[5] Chickering, D. M., Heckerman, D., and Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. Technical Report MSR-TR-97-07, Microsoft Research, Redmond, WA.

[6] Dayal, P., Trebst, S., Wessel, S., Würtz, D., Troyer, M., Sabhapandit, S., and Coppersmith, S. (2004). Performance limitations of flat histogram methods and optimality of Wang-Langdau sampling. *Physical Review Letters*, 92(9):097201.

[7] Deb, K. and Goldberg, D. E. (1991). Analyzing deception in trap functions. IlliGAL Report No. 91009, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL.

[8] Deb, K. and Goldberg, D. E. (1994). Sufficient conditions for deceptive and easy binary functions. *Annals of Mathematics and Artificial Intelligence*, 10:385–408.

[9] Fischer, K. and Hertz, J. (1991). *Spin Glasses.* Cambridge University Press, Cambridge.

[10] Goldberg, D. E. (2002). *The design of innovation: Lessons from and for competent genetic algorithms*, volume 7 of *Genetic Algorithms and Evolutionary Computation.* Kluwer Academic Publishers.

[11] Harik, G. and Lobo, F. (1999). A parameter-less genetic algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, I:258–265.

[12] Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. *Proceedings of the International Conference on Genetic Algorithms (ICGA-95)*, pages 24–31.

[13] Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1997). The compact genetic algorithm. IlliGAL Report No. 97006, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL.

[14] Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1998). The compact genetic algorithm. *Proceedings of the International Conference on Evolutionary Computation (ICEC-98)*, pages 523–528.

[15] Hartmann, A. K. (1996). Cluster-exact approximation of spin glass ground states. *Physica A*, 224:480.

[16] Hartmann, A. K. (1999). Scaling of stiffness energy for 3d $\pm j$ spin glasses. *Phys. Rev. E*, 59:84.

[17] Hartmann, A. K. and Rieger, H. (2001). *Optimization Algorithms in Physics*. Wiley-VCH, Weinheim.

[18] Larrañaga, P. and Lozano, J. A., editors (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer, Boston, MA.

[19] Lima, C. F. and Lobo, F. G. (2004). Parameter-less optimization with the extended compact genetic algorithm and iterated local search. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pages 1328–1339.

[20] Lin, T.-K. and Pelikan, M. (2004). Parameter-less compact genetic algorithm. Unpublished Technical Report, Department of Mathematics and Computer Science, University of Missouri, St. Lous.

[21] Mezard, M., Parisi, G., and Virasoro, M. (1987). *Spin glass theory and beyond*. World Scientific, Singapore.

[22] Middleton, A. and Fisher, D. S. (2002). The three-dimensional random field Ising magnet: Interfaces, scaling, and the nature of states. *Phys. Rev. B*, 65:134411.

[23] Mühlenbein, H. (1992). How genetic algorithms really work: I.Mutation and Hillclimbing. In Männer, R. and Manderick, B., editors, *Parallel Problem Solving from Nature*, pages 15–25, Amsterdam, Netherlands. Elsevier Science.

[24] Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*, pages 178–187.

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

[26] Pelikan, M. (2002). *Bayesian optimization algorithm: From single level to hierarchy*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023.

[27] Pelikan, M. (2005). *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer-Verlag.

[28] Pelikan, M. and Goldberg, D. E. (2000). Hierarchical problem solving by the Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 275–282. Also IlliGAL Report No. 2000002.

[29] Pelikan, M. and Goldberg, D. E. (2001). Escaping hierarchical traps with competent genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518. Also IlliGAL Report No. 2000020.

[30] Pelikan, M. and Goldberg, D. E. (2003a). Hierarchical boa solves ising spin glasses and maxsat. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, II:1275–1286. Also IlliGAL Report No. 2003001.

[31] Pelikan, M. and Goldberg, D. E. (2003b). A hierarchy machine: Learning to optimize from nature and humans. *Complexity*, 8(5):36–45.

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

[33] Pelikan, M. and Hartmann, A. (2006). Searching for ground states of Ising spin glasses with hierarchical BOA and cluster exact approximation. In Cantú-Paz, E., Pelikan, M., and Sastry, K., editors, *Scalable optimization via probabilistic modeling: From algorithms to applications*. Springer.

[34] Pelikan, M. and Lin, T.-K. (2004). Parameter-less hierarchical boa. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, 2:24–35.

[35] Pelikan, M. and Lobo, F. G. (1999). Parameter-less genetic algorithm: A worst-case time and space complexity analysis. IlliGAL Report No. 99014, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL.

[36] Pelikan, M., Sastry, K., and Goldberg, D. E. (2002b). Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258. Also IlliGAL Report No. 2001029.

[37] Spin Glass Ground State Server (2004). `http://www.informatik.uni-koeln.de/ls_juenger/research/sgs/sgs.html`. University of Köln, Germany.

[38] Thierens, D. (1995). *Analysis and design of genetic algorithms*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium.

[39] Thierens, D. and Goldberg, D. (1994). Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving from Nature*, pages 116–121.

[40] Thierens, D., Goldberg, D. E., and Pereira, A. G. (1998). Domino convergence, drift, and the temporal-salience structure of problems. *Proceedings of the International Conference on Evolutionary Computation (ICEC-98)*, pages 535–540.

[41] Young, A., editor (1998). *Spin glasses and random fields*. World Scientific, Singapore.