The hierarchical Bayesian optimization algorithm (hBOA) [35, 34] is an estimation of distribution algorithm (EDA) [2, 29, 26, 36]. hBOA evolves a population of candidate solutions. The population is initially generated at random according to the uniform distribution over all n-bit strings. Each iteration starts by selecting a population of promising solutions using any common selection method of genetic and evolutionary algorithms; in this paper, we use binary tournament selection. New solutions are generated by building a Bayesian network (BN) with local structures [8, 11] for the selected solutions and sampling from the probability distribution encoded by the built Bayesian network. To ensure useful-diversity maintenance, the new candidate solutions are incorporated into the original population using restricted tournament replacement (RTR) [14]. The run is terminated when some user-defined termination criteria are met; for example, the run may be terminated when a prespecified maximum number of iterations has been executed.

The basic procedure of the genetic algorithm (GA) [16, 12] variants used here is similar to that of hBOA. The only difference is the way in which the selected solutions are processed to generate new candidate solutions. In GA, instead of using BNs, new solutions are created by applying crossover and mutation to the selected solutions. Here we use two-point crossover or uniform crossover, and bit-flip mutation [12].

Incorporating local search often improves efficiency of evolutionary algorithms. Even a simple deterministic hill climber (DHC) was shown to lead to substantial speedups of hBOA and GA on finite-dimensional spin glass models [34]. That is why we decided to incorporate DHC into both hBOA and GA also in this study. DHC takes a candidate solution represented by an n-bit binary string on input. Then, it performs one-bit changes on the solution that lead to the maximum improvement of solution quality. DHC is terminated when no single-bit flip improves solution quality. Here, DHC is used to improve every solution in the population before the evaluation is performed.

3. SK SPIN GLASS

This section describes the Sherrington-Kirkpatrick (SK) spin glass, and the branch and bound algorithm.

3.1 SK Spin Glass

The Sherrington-Kirkpatrick spin glass [22] is described by a set of spins $\{s_i\}$ and a set of couplings $\{J_{i,j}\}$ between all pairs of spins. Thus, unlike in finite-dimensional spin-glass models analyzed in the context of EDAs in many previous studies [39, 17, 41, 38, 44], the SK model does not limit the range of spin-spin interactions to only neighbors in a lattice. 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 physical magnetic systems; nevertheless, the two-state Ising model comprises all basic effects also found in more realistic models of magnetic systems with more degrees of freedom.

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

$$H(C) = -\sum_{i<j} J_{i,j} s_i s_j. \quad (1)$$

In this paper the goal is to find ground states (spin configurations with the minimum possible energy) for given coupling constants. The problem of finding ground states is NP-complete even when the interactions are limited only to neighbors in a 3D lattice [3]; the SK spin glass is thus certainly NP-complete (unless we severely restrict couplings).

In order to obtain thermodynamically relevant quantities, all measurements of a spin-glass system have to be averaged over many disorder instances of random spin-spin couplings. Here we consider random instances of the SK model with couplings generated from the Gaussian distribution with zero mean and unit variance, $N(0, 1)$, which is one of the common coupling distributions [5, 7, 21, 32, 20, 25].

3.2 Branch and Bound for the SK Spin Glass

The branch-and-bound algorithm for finding ground states of SK spin-glass instances is based on a total enumeration of the space of all spin configurations. The space of spin configurations is explored by parsing a tree in which each level corresponds to one spin and the subtrees below the nodes at this level correspond to the different values this spin can obtain. To make the enumeration more efficient, branch and bound uses bounds on the energy to cut large parts of the tree, which can be proved to not lead to better solutions than the best solution found. We tried two versions of branch and bound for finding ground states of SK spin glasses. Here we outline the basic principle of the variant that performed best adopted from refs. [24, 15, 23].

The branch-and-bound algorithm for an SK spin glass of n spins $\{s_1, s_2, \ldots, s_n\}$ proceeds by iteratively finding the best configurations for the first $j = 2$ to n spins. For each value of j, the result for $(j - 1)$ is used to provide the bounds. Whenever the current branch can be shown to provide at most as good solutions as the best solution so far, the branch is cut (and not explored).

The lower bound on the energy of a reduced system with only the first j spins for any $j \leq n$, denoting the minimum energy for such a system by f_j^*, is $f_j^* \geq f_{j-1}^* - \sum_{i=1}^{j-1} |J_{i,j}|$. The last inequality holds because f_{j-1}^* is the minimum energy for the reduced system of only the first $(j-1)$ spins and the largest decrease of energy by adding s_j into the reduced system is given by the sum of the absolute values of all the couplings between s_j and the spins in the reduced system.

To make the branch and bound faster, we execute several runs of a stochastic hill climbing to provide a good starting point for each reduced problem, allowing more branches to be cut. With the described approach, ground states of instances of up to about 90 spins can be found in practical time on a reasonable sequential computer.

4. INITIAL EXPERIMENTS

This section describes initial experiments. As the first step, we generated a large number of random SK instances of sizes up to $n = 80$ spins and applied the branch and bound to find the ground states of these instances. Next, hBOA was applied to all these problem instances and the performance and parameters of hBOA were analyzed.

4.1 Initial Set of Instances of up to 80 Spins

First, we generated 10^4 SK ground-state instances for each problem size from $n = 20$ spins to $n = 80$ spins with step 2. Branch and bound was applied to each of these instances to determine the true ground state, providing us with a set of 310,000 unique problem instances of different sizes with known ground states. The motivation for using so many instances for each problem size and for increasing the problem size with the step of only 2 was that the problem difficulty varies significantly across the different instances and, as a result, it was desirable to use as many problem instances as possible.

4.2 hBOA Parameters, Experimental Setup

To represent spin configurations of n spins, hBOA uses an n-bit binary string where the i-th bit determines the state of the spin s_i; -1 is represented with a 0, $+1$ is represented with a 1. Each candidate solution is assigned the fitness, which is equal to the negative energy of the configuration. Thus, maximizing fitness corresponds to minimizing energy.

Some hBOA parameters do not depend much on the problem instance being solved, and that is why they are typically set to some default values, which were shown to perform well across a broad range of problems. To select promising solutions, we use binary tournament selection. New solutions are incorporated into the original population using restricted tournament replacement with window size $w = \max\{n, N/5\}$ where n is the number of bits in a solution and N is the population size. Bayesian networks are selected based on the Bayesian-Dirichlet metric with likelihood equivalence [9, 8], which is extended with a penalty for model complexity [11, 37, 34]. The complexity of Bayesian networks used in hBOA was not directly restricted.

The best values of two hBOA parameters critically depend on the problem being solved: (1) the population size and (2) the maximum number of iterations. The maximum number of iterations is typically set to be proportional to the number of bits in the problem, which is supported by the domino convergence model for exponentially scaled problems [47]. Since from some preliminary experiments it was clear that the number of iterations would be very small for all problems of $n \leq 80$, typically less than 10 even for $n = 80$, we set the number of iterations to the number of bits in the problem. Experiments confirmed that the used bound on the number of iterations was certainly sufficient.

To set the population size, we have used the bisection method [42, 34], which automatically determines the necessary population size for reliable convergence to the optimum in 10 out of 10 independent runs. This is done for each problem instance so that the resulting population sizes are as small as possible, which typically minimizes the execution time. Each run in the bisection is terminated either when the global optimum (ground state) has been reached (success), or when the maximum number of iterations has been exceeded without finding the global optimum (failure).

4.3 Analysis of hBOA for up to 80 Spins

For each problem instance, after determining an adequate population size with bisection and making 10 independent runs of hBOA with that population size, we record four important statistics for these 10 successful runs: (1) the population size, (2) the number of iterations, (3) the number of evaluations, and (4) the number of single-bit flips of the local searcher. For each problem size, we thus consider 100,000 successful hBOA runs, yielding a total of 3,100,000 successful hBOA runs for problems 20 to 80 bits. In order to solve larger problems, especially the results for the population size and the number of iterations are useful. On the other hand, to analyze the time complexity of hBOA, the number of evaluations and the number of flips are most important.

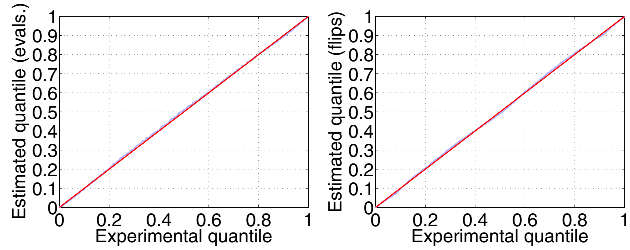

Figure 1: Comparison of the experimental distribution (the number of evaluations, and the number of flips) and the corresponding log-normal distribution estimates for $n = 80$.

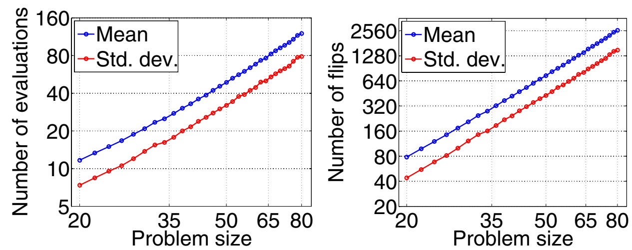

Figure 2: Mean and standard deviation of the log-normal approximation for the number of evaluations and the number of flips for $n = 20$ to 80.

The first step in analyzing the results of hBOA is to identify the probability distribution that the different observed statistics follow. By identifying a specific distribution type, the results of the analysis should be much more accurate and practically useful. Based on our prior work in spin glasses and preliminary experiments, there are two distributions that might be applicable: (1) the log-normal distribution and (2) the generalized extreme value distribution. For all aforementioned statistics we first estimate the parameters of both the distributions and then compare these estimates to the underlying data. For $n = 20$ to 80, the most stable results are obtained with the log-normal distribution and that is why we decided to use log-normal distributions in this and further analyses.

Figure 1 illustrates the match between the estimated log-normal distribution and the experimental data for the number of evaluations and the number of flips for $n = 80$. Analogous matches were found for smaller problem sizes. Due to space limitations, we only show the match between the cumulative density of the observed data and the distribution estimates, and we omit the results for the population size and the number of iterations.

Figure 2 shows the mean and standard deviation of the distribution estimates for the entire range of SK instances from $n = 20$ to $n = 80$ with step 2. Due to space limitations, we only show the estimates for the number of evaluations and the number of flips, which are relevant for the time complexity of hBOA (all remaining results will be provided online). Nonetheless, it is important to not only study the mean statistics, but also to analyze the tails of the estimated distributions. This is especially necessary for problems like mean-field SK spin glasses for which the difficulty of problem instances varies significantly and, as a result, while many instances are relatively easy, solving the most difficult instances becomes especially challenging.

5. HOW TO LOCATE GLOBAL OPTIMA RELIABLY FOR BIGGER PROBLEMS?

To make sure that hBOA finds a ground state reliably, it is necessary to set the population size and the maximum number of iterations to sufficiently large values. The larger the population size and the number of iterations, the more likely hBOA finds the optimum. Of course, as the problem size increases, the population-sizing and time-to-convergence requirements will increase as well, just like was indicated by the initial results presented in the previous section.

In this section we present three approaches to reliably locate ground states of SK instances unsolvable with the branch-and-bound algorithm. The first two approaches are based on the statistical models of the population size and the number of iterations for smaller problem instances, such as those developed in the previous section. On the other hand, the last approach does not require any statistical model or prior experiments, although it still requires an estimate of the upper bound on the maximum number of iterations. The proposed approaches are not limited to the SK spin-glass model and can thus be used to reliably identify the global optima of other difficult problems.

5.1 Modeling the Percentiles

The first approach models the growth of the percentiles of the estimated probability distributions. As the input, we use the estimated distributions of the population size and the number of iterations for SK instances of $n \leq 80$ spins.

For each problem size n, we first compute the 99.999 percentile of the population-size model so that it is ensured that the resulting population sizes will be sufficiently large for all but the 0.001% most difficult instances. Then, we approximate the growth of the 99.999 percentile and use this approximation to predict sufficient population sizes for larger problems. Since the estimation of the growth function is also subject to error, we can use a 95% confidence bound for the new predictions and choose the upper bound given by this confidence bound. An analogous approach can then be used for the number of iterations.

Of course, the two confidence bounds involved in this approach can be changed and the estimation should be applicable regardless of the model used to fit the distribution of the population size and the number of iterations; nonetheless, it is important to build an accurate approximation of the true distribution of the population sizes in order to have an accurate enough approximation of the chosen percentile. Furthermore, it is important to use an appropriate growth function to predict the percentiles in order to minimize the errors for predicting the parameters for bigger problems.

The same approach can be used to predict the population size and the number of iterations for other population-based evolutionary algorithms, such as the genetic algorithm. Furthermore, when applied to other types of stochastic optimization algorithms, other parameters may be modeled accordingly. For example, if we were using simulated annealing, we could model the rate of the temperature decrease.

Figure 3 shows the percentiles for the population size and the number of iterations obtained from the log-normal distribution estimates for $n \leq 80$ presented in the previous section, and the best-fit curves estimating the growth of these percentiles created with the Matlab curve-fitting toolbox. Best approximation of the growth of both the quantities is obtained with a power-law fit of the form $an^b + c$ where n

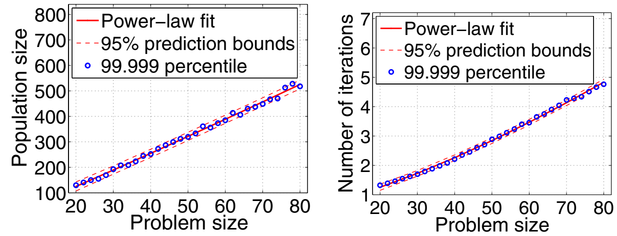

Figure 3: A model of the population size and the number of iterations based on the 99.999 percentile of the estimated distributions for $n \leq 80$.

is the number of bits (spins). The estimated parameters of the best power-law fit of the population-size percentiles follow (adjusted R^2 of the estimate is 0.9963): $a = 5.094$, $b = 1.056$, and $c = 4.476$. The estimated parameters of the best power-law fit for the number of iterations follow (adjusted R^2 is 0.9983): $a = 0.01109$, $b = 1.356$, and $c = 0.6109$.

The model can be extended to larger problem sizes (we omit the figures due to space limitations). For example, the 95%-confidence upper bound on the population size for $n = 200$ is about 1505, whereas the upper bound on the number of iterations for $n = 200$ is about 16.

5.2 Modeling the Distribution Parameters

The basic idea of this approach to estimating an adequate population size and the maximum number of iterations is to directly model the distribution parameters and then predict the distribution parameters for larger problems. Based on the predicted probability distributions of the population size and the number of iterations, we can predict adequate values of these parameters to solve at least a specified percentage of larger problem instances (with an arbitrary probability of error).

Specifically, we start with the estimated mean and deviation of the underlying normal distribution for the log-normal fit of the population size and the number of iterations. The growth of the mean and the standard deviation is then approximated using an appropriate function to fit the two estimated statistics.

Figure 4 shows the estimated distribution parameters and the power-law fit for these parameters obtained with the Matlab curve-fitting toolbox. For both the mean and the standard deviation, the best match is obtained with the power-law fit. For the mean, the R^2 for the fit is 0.9998, whereas for the standard deviation, the R^2 is 0.7879. Thus, for the standard deviation, the fit is not very accurate. The main reason for this is that the standard deviation of the underlying normal distribution is rather noisy and it is difficult to find a model that fits the standard deviation estimates accurately. Therefore, it appears that the first approach to predicting the population size and the number of iterations for larger problems results in more accurate estimates, although both approaches yield comparable predictions.

5.3 Population Doubling

One of the main features of the problem of finding ground states of various spin-glass models, including those in two and three dimensions, is that the problem difficulty varies significantly between different problem instances. As a result, even the population size and the number of iterations vary considerably between the different problem instances. Estimating an upper bound for the population size and the

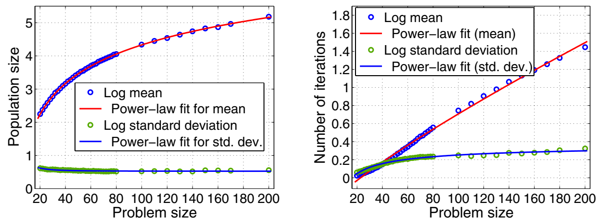

Figure 4: A model of the parameters of the probability distributions governing hBOA parameters.

number of iterations enables us to guarantee that with high probability, the found spin configurations will indeed be ground states. Nonetheless, since many problem instances are significantly simpler than the predicted worst case, this causes us to waste computational resources on the simple problems. Furthermore, while in some cases the distribution of the parameters may be relatively straightforward to estimate accurately, these estimates may be misleading or difficult to obtain in other cases.

One way to circumvent these problems is to use the following approach, which is loosely based on the parameter-less genetic algorithm [13] and the greedy population sizing [45]. The approach starts with a relatively small population size N_{init}, and executes num_{runs} hBOA runs with that population size (for example, $num_{\text{runs}} = 10$) and a sufficient upper bound on the number of iterations. The best solution of each run is recorded. Next, the procedure is repeated with the double population size $2N_{\text{init}}$, and again the best solution of each run is recorded. The doubling continues until it seems unnecessary to further increase the population size, and the best solution found is then returned.

If a certain population size is too small to reliably identify the global optimum, we can expect two things to happen: (1) different runs would result in solutions of different quality (at least some of these solutions would thus be only locally optimal), and (2) doubling the population size would provide better solutions. Based on these observations, we decided to terminate the population-doubling procedure if and only if all num_{runs} runs end up in the solution of the same quality and the solution has not improved for more than max_{failures} rounds of population doubling (for example, $max_{\text{failures}} = 2$). Of course, the method can be tuned by changing the parameters num_{runs} and max_{failures}, depending on whether the primary target is reliability or efficiency. To improve performance further (at the expense of reliability), the termination criterion can be further relaxed by not requiring all runs to find solutions of the same quality.

There are two main advantages of the above procedure for discovering the global optima. First of all, unlike the previous two approaches, here simpler problems will indeed be expected to use less computational resources. Second, we do not have to provide any parameter estimates except for the maximum number of iterations, which is typically easy to estimate sufficiently well. Furthermore, even if we do not know how to properly upper bound the maximum number of iterations, we can use other common termination criteria. For example, each run can be terminated when the fitness of the best solution does not improve for a specified number of iterations or when the best fitness obtained is almost equal to the average fitness of the population. Since in many cases it is difficult to estimate the growth of the population size, the algorithm presented in this section may be the only feasible approach out of the three approaches discussed in this paper.

Clearly, there are also disadvantages: Most importantly, if we have an accurate enough statistical model for the population size and the number of iterations, modeling the percentiles or parameters of these distributions allows a highly reliable detection of global optima for larger problems. On the other hand, if we use the approach based on doubling the population size, although the termination criteria are designed to yield reliable results, there are no guarantees that we indeed locate the global optimum.

6. EXPERIMENTS ON LARGER PROBLEM INSTANCES

This section shows the results of applying hBOA to SK instances of up to $n = 300$ spins. The problems of sizes $n \in [100, 200]$ were solved using parameter estimates created by statistical models of the percentiles of the estimated distributions of these parameters. Then, the results on problems of size $n \leq 200$ were used to improve the model of the population size and the number of iterations, which was then used to estimate adequate values of the population size and the number of iterations for $n = 300$.

6.1 Solving Instances of 100 to 200 Spins

To solve larger SK spin-glass instances, we first generate 1000 instances for $n = 100$ to $n = 200$ spins with step 10; the number of instances for each problem size is decreased because as the problem size grows, it becomes more difficult to reliably locate the ground states. Then, we use the model of the growth of the 99.999 percentile of the population size and the number of iterations to estimate adequate values of these parameters for each value of n with confidence 95%. To further improve reliability, for each problem instance we perform 10 independent runs and record the best solution found. Since for $n \geq 100$ we can no longer use the branch and bound to identify true ground states and hBOA is not guaranteed to always find the global optimum, we verified all results with the population-doubling approach and hysteretic optimization [31], as is discussed later (see section 6.2). The additional simulations confirmed all the results.

After determining the ground states of spin-glass instances for $n = 100$ to $n = 200$, we use bisection to find the optimal population size for hBOA on each instance, similarly as done in the experiments for $n \leq 80$ (see section 4). Since for each problem size we generate only 1000 random instances, in order to obtain more results, we repeat bisection 10 times for each instance, always with different initial parameters and a different random seed. Therefore, for each problem instance, we end up with 100 successful runs (10 successful hBOA runs for each of the 10 bisection runs), and the overall number of successful runs for each problem size is 100,000. Overall, for problem sizes $n = 100$ to $n = 200$, we performed 1,100,000 successful runs with hBOA.

Analogously to the results on SK instances of sizes $n \leq 80$, we fit the population size, the number of iterations, the number of evaluations, and the number of flips for each problem size using log-normal distribution. The resulting mean and standard deviation for all the statistics is shown in figure 5.

An analogous fit to the 99.999 percentile of the population size and the number of iterations is obtained for problems

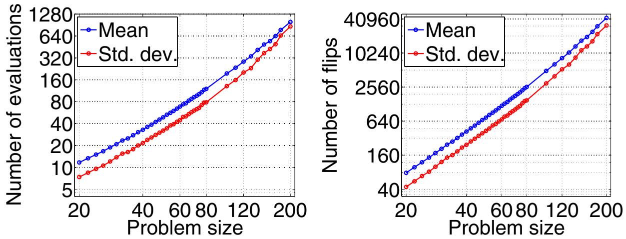

Figure 5: Mean and standard deviation of the log-normal approximation for the number of evaluations and the number of flips for $n = 20$ to 200.

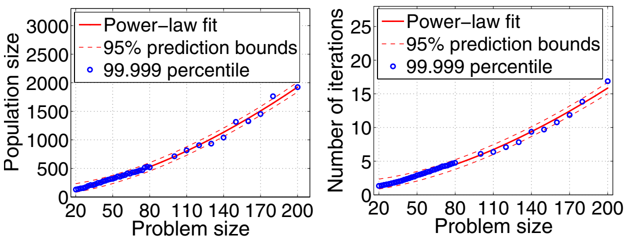

Figure 6: A model of the population size and the number of iterations based on the 99.999 percentile of the estimated distributions for $n \leq 200$.

of sizes $n \leq 200$ with step 2 from $n = 20$ to $n = 80$ and with step 10 from $n = 100$ to $n = 200$. The fit is shown in figure 6 (left-hand side). The power-law fit performed the best, resulting in the following parameters for the population size (R^2 value for the fit is 0.9938): $a = 0.3582$, $b = 1.61$, and $c = 113.3$. The power-law fit for the number of iterations had the following parameters (R^2 value for the fit is 0.9909): $a = 0.002379$, $b = 1.646$, and $c = 1.264$.

We use the above model to predict adequate values of the population size and the number of iterations for $n = 300$, 3812.44 for the population size and 30.1702 for the number of iterations, respectively.

6.2 Solving Instances of 300 Spins

To solve even larger problem instances, we generate 1000 random SK instances of $n = 300$ spins. These instances are then solved using hBOA with the population size set according to the upper limit estimated in the previous section. To improve reliability, to locate the ground states for $n = 300$, we use the population size of 4,800 and the bound on the number of iterations of 300. Similarly as in the experiments for $n \in [100, 200]$, hBOA is run 10 times on each problem instance and the best solution found in these 10 runs is recorded. Then, to determine optimal hBOA performance, we run bisection 10 times on each problem instance.

The running time for hBOA with optimal population size for $n = 300$ varied significantly—it ranged from 0.9 seconds to 2518.8 seconds (with the average of 40.4 seconds) on a 3 GHz Intel Xeon processor. The running times with the predicted upper bound on the population size and the number of iterations were much more stable since the same parameters were used for all instances; these runs took approximately 12,148 seconds on average.

After analyzing the distribution of the various statistics collected from hBOA runs on spin-glass instances of $n = 300$ spins, it became clear that while for smaller problems, the log-normal distribution provided an accurate and stable model of the true distribution, for $n = 300$, the fit with the log-normal distribution does no longer seem to be the best option and the distribution is more accurately reflected by the generalized extremal value distribution. This is surprising, since the log-normal fits for smaller problems are very accurate and they provide more stable results than the generalized extreme value fits. As presented in reference [4] the thermodynamic limiting behavior of the SK model is only probed for $n \gtrsim 150$ spins. Interestingly, this threshold agrees qualitatively with the change of the fitting functions.

To verify the ground states obtained with the parameter estimates based on the log-normal distribution, we applied the population-doubling scheme to all instances of $n \in [100, 300]$ spins. Additionally, we implemented hysteretic optimization (HO) [31], which is known to perform well on fully connected SK spin glasses. HO was run on each SK instance several tens of times for a range of parameter settings and initial random seeds. The additional simulations confirmed all the results for SK instances of $n \in [100, 300]$ spins.

7. COMPARISON OF HBOA AND GA

This section compares the performance of hBOA, GA with two-point crossover and bit-flip mutation, and GA with uniform crossover and bit-flip mutation. All algorithms are configured in the same way except for the population size, which is obtained separately for each problem instance and each algorithm using the bisection method described earlier. Since the experiments were carried out on a number of different parallel computers, no comparisons were done for the overall execution time (this remains for future work).

To compare algorithms 'A' and 'B,' we first compute the ratio of the number of evaluations (flips) required by A and B, separately for each problem instance. Then, we average these ratios over all instances of the same problem size. If the ratio is greater than 1, the algorithm B requires fewer evaluations (flips) than the algorithm A; therefore, with respect to the number of fitness evaluations (flips), we can conclude that B is better than A. Similarly, if the ratio is smaller than 1, then we can conclude that with respect to the number of evaluations, A performs better than B.

The results of pair-wise comparisons between all three algorithms are shown in figure 7. The results clearly indicate that hBOA outperforms both GA variants and the gap between hBOA and the GA variants increases with problem size. Therefore, for larger problems, the performance differences can be expected to grow further. From the comparison of the two GA variants, it is clear that while with respect to the number of evaluations, two-point crossover performs better, with respect to the number of flips, uniform crossover performs better with increasing problem size.

While the differences between hBOA and GA are not as significant for problem sizes considered in this work, since the gap between these algorithms grows with problem size, for much larger problem, the differences can be expected to become significant enough to make GA variants intractable on problems solvable with hBOA in practical time.

8. FUTURE WORK

The most immediate milestone to tackle is to further increase the size of the systems for which we can reliably identify ground states with the ultimate goal of obtaining global optima for SK instances significantly larger than 10^3 spins. To succeed in this goal, the techniques developed in this pa-

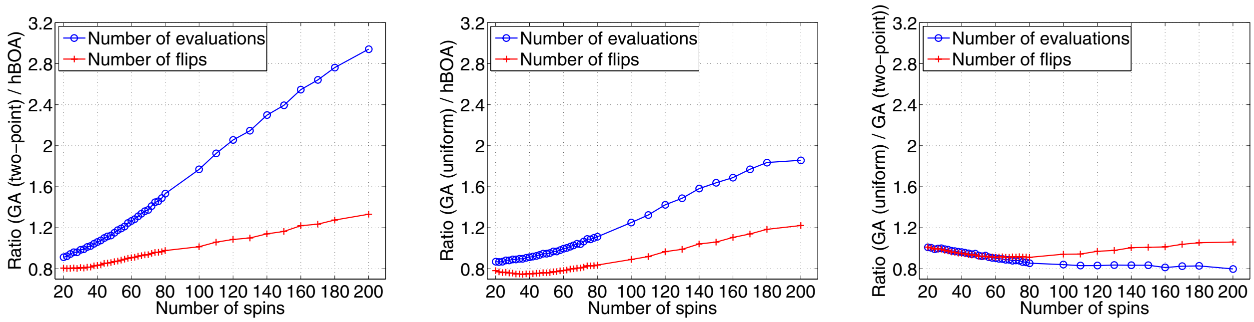

Figure 7: Comparison of hBOA, GA with uniform crossover, and GA with two-point crossover with respect to the overall number of evaluations and the number of flips of the local searcher. The relative performance is visualized as the ratio of the number of evaluations (number of flips) required by pairs of compared algorithms.

per may be used in combination with additional efficiency enhancement techniques [43, 40], especially hybridization.

Another interesting topic for future work is to use problem instances obtained in this work to test other optimization algorithms and compare their performance with that of hBOA and GA. Good candidates for such a comparison are parallel tempering [18, 28, 20, 25], GA with triadic crossover [30], extremal optimization [6], and hysteretic optimization [31]. All these algorithms were argued to solve relatively large instances of various classes of the SK spin glass.

Finally, this work can be extended to other interesting types of instances of the SK spin-glass model and other difficult combinatorial problems. One of the important extensions is to consider bimodal distributions, which are of interest especially due to the high degeneracy of the ground state [28]. Another interesting extension is to impose a distance metric between pairs of spins and modify the coupling distribution based on the distance between the two connected spins [19]. The latter has the advantage that, while the connectivity of the model is kept constant, the range of the interactions can be tuned continuously between a system in a mean-field universality class to a short-range nearest-neighbor model. This provides an ideal benchmark for optimization algorithms in general.

9. SUMMARY AND CONCLUSIONS

This paper applied the hierarchical Bayesian optimization algorithm (hBOA) and the genetic algorithm (GA) to the problem of finding ground states of instances of the Sherrington-Kirkpatrick (SK) spin-glass model with Ising spins and Gaussian couplings, and analyzed performance of these algorithms on a large set of instances of the SK model. First, 10,000 random problem instances were generated for each problem size from $n = 20$ to $n = 80$ with step 2 and ground states of all generated instances were determined using the branch-and-bound algorithm. Then, hBOA was applied to these instances, and its parameters and performance were analyzed in detail. Since problems of $n \geq 100$ spins are intractable with branch and bound, we proposed several approaches to reliably identifying ground states of such problem instances with hBOA. One of the proposed approaches was applied to problem instances of sizes $n \in [100, 300]$ (1000 random instances for each problem size). Analogous experiments as with hBOA were also performed with the genetic algorithm (GA) with bit-flip mutation and two common crossover operators. Performance of hBOA and the two GA variants was compared, indicating that hBOA outperforms both GA variants and the gap between these two algorithms increases with problem size.

Our study presents for the first time a detailed study of genetic and evolutionary algorithms applied to the problem of finding ground states of the SK spin-glass model. The lessons learned and the techniques developed in tackling this challenge should be important for optimization researchers as well as practitioners.

Acknowledgments

The authors would like to thank Károly F. Pál for help with the implementation of hysteretic optimization. This project was sponsored by the National Science Foundation under CAREER grant ECS-0547013, by the Air Force Office of Scientific Research, USAF, under grant FA9550-06-1-0096, and by the University of Missouri in St. Louis through the High Performance Computing Collaboratory sponsored by Information Technology Services, and the Research Award and Research Board programs. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Air Force Office of Scientific Research, or the U.S. Government. Some experiments were done using the hBOA software developed by M. Pelikan and D. E. Goldberg at the Univ. of Illinois at Urbana-Champaign and most experiments were performed on the Beowulf cluster maintained by ITS at the Univ. of Missouri in St. Louis. H.G.K. would like to thank the Swiss National Science Foundation for financial support under grant No. PP002-114713. Some simulations were performed on the Gonzales cluster of ETH Zürich.

10. REFERENCES

[1] T. Aspelmeier, M. A. Moore, and A. P. Young. Interface energies in Ising spin glasses. *Phys. Rev. Lett.*, 90:127202, 2003.

[2] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Tech. Rep. No. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.

[3] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical, Nuclear and General*, 15(10):3241–3253, 1982.

[4] A. Billoire. Some aspects of infinite range models of spin glasses: theory and numerical simulations. (arXiv:cond-mat/0709.1552), 2007.

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

[6] S. Boettcher. Extremal optimization for Sherrington-Kirkpatrick spin glasses. *Eur. Phys. J. B*, 46:501–505, 2005.

[7] J.-P. Bouchaud, F. Krzakala, and O. C. Martin. Energy exponents and corrections to scaling in Ising spin glasses. *Phys. Rev. B*, 68:224404, 2003.

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

[9] G. F. Cooper and E. H. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[10] A. Crisanti, G. Paladin, and H.-J. S. A. Vulpiani. Replica trick and fluctuations in disordered systems. *J. Phys. I*, 2:1325–1332, 1992.

[11] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In M. I. Jordan, editor, *Graphical models*, pages 421–459. MIT Press, Cambridge, MA, 1999.

[12] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

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

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

[15] A. Hartwig, F. Daske, and S. Kobe. A recursive branch-and-bound algorithm for the exact ground state of Ising spin-glass models. *Computer Physics Communications*, 32:133–138, 1984.

[16] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[17] R. Höns. *Estimation of Distribution Algorithms and Minimum Relative Entropy*. PhD thesis, University of Bonn, Germany, 2005.

[18] K. Hukushima and K. Nemoto. Exchange Monte Carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.*, 65:1604, 1996.

[19] H. G. Katzgraber. Spin glasses and algorithm benchmarks: A one-dimensional view. In *Proc. of the International Workshop on Statistical-Mechanical Informatics*, 2007.

[20] H. G. Katzgraber, M. Körner, F. Liers, M. Jünger, and A. K. Hartmann. Universality-class dependence of energy distributions in spin glasses. *Phys. Rev. B*, 72:094421, 2005.

[21] N. Kawashima and H. Rieger. Recent Progress in Spin Glasses. 2003. (cond-mat/0312432).

[22] S. Kirkpatrick and D. Sherrington. Infinite-ranged models of spin-glasses. *Phys. Rev. B*, 17(11):4384–4403, Jun 1978.

[23] S. Kobe. Ground-state energy and frustration of the Sherrington-Kirkpatrick model and related models. ArXiv Condensed Matter e-print cond-mat/03116570, University of Dresden, 2003.

[24] S. Kobe and A. Hartwig. Exact ground state of finite amorphous Ising systems. *Computer Physics Communications*, 16:1–4, 1978.

[25] M. Körner, H. G. Katzgraber, and A. K. Hartmann. Probing tails of energy distributions using importance-sampling in the disorder with a guiding function. *J. Stat. Mech.*, P04005, 2006.

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

[27] M. Mézard, G. Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, Singapore, 1987.

[28] J. J. Moreno, H. G. Katzgraber, and A. K. Hartmann. Finding low-temperature states with parallel tempering, simulated annealing and simple Monte Carlo. *Int. J. Mod. Phys. C*, 14:285, 2003.

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

[30] K. F. Pál. The ground state energy of the Edwards-Anderson Ising spin glass with a hybrid genetic algorithm. *Physica A*, 223(3-4):283–292, 1996.

[31] K. F. Pál. Hysteretic optimization for the Sherrington Kirkpatrick spin glass. *Physica A*, 367:261–268, 2006.

[32] M. Palassini. Ground-state energy fluctuations in the Sherrington-Kirkpatrick model. 2003. (cond-mat/0307713).

[33] G. Parisi. Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.*, 43:1754, 1979.

[34] M. Pelikan. *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer, 2005.

[35] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518, 2001.

[36] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[37] M. Pelikan, D. E. Goldberg, and K. Sastry. Bayesian optimization algorithm, decision graphs, and Occam's razor. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 519–526, 2001.

[38] M. Pelikan and A. K. Hartmann. Searching for ground states of Ising spin glasses with hierarchical BOA and cluster exact approximation. In E. Cantú-Paz, M. Pelikan, and K. Sastry, editors, *Scalable optimization via probabilistic modeling: From algorithms to applications*, pages 333–349. Springer, 2006.

[39] M. Pelikan, J. Ocenasek, S. Trebst, M. Troyer, and F. Alet. Computational complexity and simulation of rare events of ising spin glasses. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, 2:36–47, 2004.

[40] M. Pelikan, K. Sastry, and D. E. Goldberg. Sporadic model building for efficiency enhancement of hBOA. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, 2006.

[41] R. Santana. Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation*, 13(1):67–97, 2005.

[42] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, Department of General Engineering, Urbana, IL, 2001.

[43] K. Sastry, M. Pelikan, and D. E. Goldberg. Efficiency enhancement of estimation of distribution algorithms. In M. Pelikan, K. Sastry, and E. Cantú-Paz, editors, *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, pages 161–185. Springer, 2006.

[44] S. K. Shakya, J. A. McCall, and D. F. Brown. Solving the Ising spin glass problem using a bivariate EDA based on Markov random fields. *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2006)*, pages 908– 915, 2006.

[45] E. A. Smorodkina and D. R. Tauritz. Greedy population sizing for evolutionary algorithms. *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 2181–2187, 2007.

[46] M. Talagrand. The Parisi formula. *Ann. of Math.*, 163:221, 2006.

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