Hierarchical BOA Solves Ising Spin Glasses and MAXSAT

Martin Pelikan and David E. Goldberg

IlliGAL Report No. 2003001
January 2003

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-2346
Fax: (217) 244-5705

Hierarchical BOA Solves Ising Spin Glasses and MAXSAT

Martin Pelikan
Computational Laboratory (Colab)
Swiss Federal Institute of Technology (ETH)
Hirschengraben 84, 8052 Zurich, Switzerland
pelikanm@inf.ethz.ch

David E. Goldberg
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
104 S. Mathews Avenue, Urbana, IL 61801
deg@illigal.ge.uiuc.edu

January 26, 2003

Abstract

Theoretical and empirical evidence exists that the hierarchical Bayesian optimization algorithm (hBOA) can solve challenging hierarchical problems and anything easier. This paper applies hBOA to two important classes of real-world problems: Ising spin-glass systems and maximum satisfiability (MAXSAT). The paper shows how easy it is to apply hBOA to real-world optimization problems. The results indicate that hBOA is capable of solving enormously difficult problems that cannot be solved by other optimizers and still provide competitive or better performance than problem-specific approaches on other problems. The results thus confirm that hBOA is a practical, robust, and scalable technique for solving challenging real-world problems.

1 Introduction

Recently, the hierarchical Bayesian optimization algorithm (hBOA) has been proposed to solve hierarchical and nearly decomposable problems (Pelikan & Goldberg, 2001; Pelikan, 2002; Goldberg, 2002). The success in designing a competent hierarchical optimizer has two important consequences. First, many complex real-world systems can be decomposed into a hierarchy (Simon, 1968), so we can expect hBOA to provide robust and scalable solution to many real-world problems. Second, many difficult hierarchical problems are intractable by any other algorithm (Pelikan, 2002) and thus hBOA should allow us to solve problems that could not be solved before.

This paper applies hBOA to two important classes of real-world problems to confirm that hierarchical decomposition is a useful concept in solving real-world problems. Two classes of problems are considered: (1) two- and three-dimensional Ising spin glass systems with periodic boundary conditions, and (2) maximum satisfiability of predicate calculus formulas in conjunctive normal form (MAXSAT). The paper shows how easy it is to apply hBOA to combinatorial problems and achieve competitive or better performance than problem-specific approaches.

The paper starts with a brief description of hBOA. Section 3 defines the problem of finding a ground state of an Ising spin glass system and presents the results of applying hBOA with and without local search to this class of problems. Section 4 defines MAXSAT and discusses its difficulty. hBOA is again combined with local search and applied to several benchmark instances of MAXSAT. For both problem classes, the performance of hBOA is compared to that of problem-specific approaches. Finally, Section 5 summarizes and concludes the paper.

2 Hierarchical Bayesian optimization algorithm (hBOA)

Probabilistic model-building genetic algorithms (PMBGAs) (Pelikan, Goldberg, & Lobo, 2002) replace traditional variation operators of genetic and evolutionary algorithms—such as crossover and mutation—by building and sampling a probabilistic model of promising solutions. PMBGAs are sometimes referred to as estimation of distribution algorithms (EDAs) (Mühlenbein & Paaß, 1996), or iterated density estimation algorithms (IDEAs) (Bosman & Thierens, 2000). For an overview of PMBGAs, see Pelikan (2002), Pelikan, Goldberg, and Lobo (2002), and Larranaga and Lozano (2002). One of the most advanced PMBGAs for discrete representations is the hierarchical Bayesian optimization algorithm (hBOA) (Pelikan & Goldberg, 2001; Pelikan, 2002; Goldberg, 2002).

hBOA evolves a population of candidate solutions to the given optimization problem starting with a random population. In each iteration, hBOA updates the population in the following four steps. hBOA first selects a population of promising solutions from the current population using one of the popular selection operators, such as tournament or truncation selection. Next, hBOA builds a Bayesian network with local structures as a probabilistic model of promising solutions. New candidate solutions are then generated by sampling the learned network. Finally, restricted tournament replacement (Harik, 1994; Pelikan & Goldberg, 2001) is used to incorporate the new solutions into the current population.

hBOA is capable of automatic discovery and exploitation of hierarchical problem decomposition, which simplifies the problem via decomposition over one or more levels. This makes hBOA capable of solving challenging hierarchical problems that are practically unsolvable by traditional black-box optimizers such as simulated annealing or traditional genetic algorithms (GAs). Of course, hBOA can solve anything easier. For more details on hBOA, see Pelikan and Goldberg (2001) and Pelikan (2002).

3 Ising spin glasses

The task of finding the ground state of an Ising spin-glass system is a well known problem of statistical physics. In context of GAs, Ising spin-glass systems are usually studied because of their interesting properties, such as symmetry and a large number of plateaus (Pelikan, Goldberg, & Cantú-Paz, 2000; Pelikan & Mühlenbein, 1999; Naudts & Naudts, 1998; Van Hoyweghen, 2001; Mühlenbein, Mahnig, & Rodriguez, 1999).

The physical state of an Ising spin-glass system is defined by (1) a set of spins $(\sigma_0, \sigma_1, \ldots, \sigma_{n-1})$, where each spin σ_i can obtain a value from $\{+1, -1\}$, and (2) a set of coupling constants J_{ij} relating pairs of spins σ_i and σ_j. A Hamiltonian specifies the energy of the system as

$$H(\sigma) = -\sum_{i,j=0}^{n} J_{ij}\sigma_i\sigma_j.$$

The task is to find a state of spins called the *ground state* for given coupling constants J_{ij} that *minimizes* the energy of the system.

The problem of finding ground states of Ising spin-glasses is equivalent to a well known combinatorial problem called *minimum-weight cut* (MIN-CUT). Since MIN-CUT is NP-complete (Monien & Sudborough, 1988), the task of finding the ground state of an unconstrained Ising spin-glass system is NP-complete—that means that exists no algorithm for solving general Ising spin glasses in polynomial time and it is believed that it's impossible to do this. Here we consider a special case, where the spins are arranged on a two- or three-dimensional grid and each spin interacts with only its nearest neighbors in the grid. Periodic boundary conditions are used to approximate the behavior of a large-scale system. Therefore, spins are arranged on a two- or three-dimensional toroid. Additionally, we constrain coupling constants to contain only two values, $J_{ij} \in \{+1, -1\}$. In the two-dimensional case, several algorithms exist that can solve the restricted class of spin glasses in polynomial time. We will compare the best known algorithms to hBOA later in this section. However, none of these methods is applicable in the three-dimensional case. In fact, the three-dimensional case with coupling constants from $\{-1, 0, +1\}$ has been shown to be NP-complete (Barahona, 1982).

3.1 Methodology

In hBOA, each state of the system is represented by an n-bit binary string, where n is the total number of spins. Each bit in a solution string determines the state of the corresponding spin: 0 denotes the state -1, 1 denotes the state $+1$. To estimate the scalability of hBOA on two-dimensional spin-glass systems, we tested the algorithm on a number of two-dimensional spin-glass systems of size from $n = 10 \times 10 = 100$ spins to $n = 16 \times 16 = 256$ spins. For each problem size, we generated 8 random problem instances with uniform distribution over all problem instances. To ensure that a correct ground state was found for each system, we verified the results using the Spin Glass Ground State Server provided by the group of Prof. Michael Jünger (http://www.informatik.uni-koeln.de/ls_juenger/projects/sgs.html).

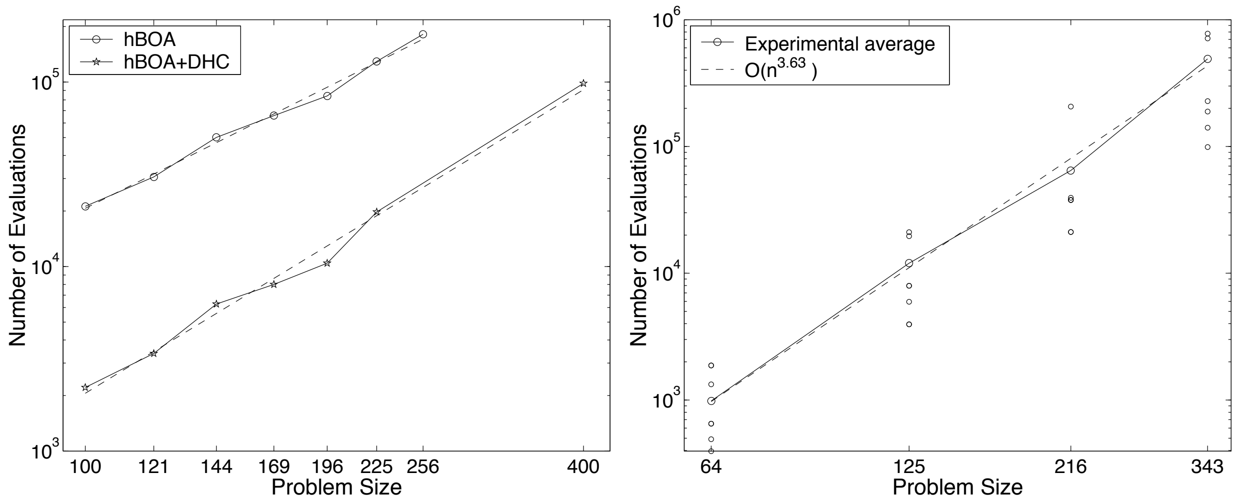
For each problem instance, 30 independent runs are performed and hBOA is required to find the optimum in all the 30 runs. The performance of hBOA is measured by the average number of evaluations until the optimum is found. The population size for each problem instance is determined empirically as the minimal population size for the algorithm to find the optimum in all the runs. A parameter-less population sizing scheme (Harik & Lobo, 1999) could be used to eliminate the need for specifying the population size in advance, which could increase the total number of evaluations by at most a logarithmic factor (Pelikan & Lobo, 1999). Binary tournament selection with replacement is used in all experiments and the window size for RTR is set to the number of bits (spins) in a problem. Bayesian networks with decision graphs are used and K2 metric with the term penalizing complex models is used to measure the quality of each candidate model.

The performance of hBOA is compared to that of hBOA combined with local search referred to as the *discrete hill climber* or DHC. DHC is applied prior to evaluation of each solution by flipping a bit that improves the solution the most; this is repeated until no more improvement is possible. For most constraint satisfaction problems including Ising spin glasses, DHC increases the computational cost of each evaluation at most n times; in practice, the increase in computational complexity is still significantly lower, because only a few bits are flipped on average.

Using local search usually improves the performance of selectorecombinative search, because the search can focus on local optima, which reveal more information about the problem than randomly generated solutions do. Furthermore, selectorecombinative search can focus its exploration on basins of attraction (peaks around each local optimum) as opposed to individual solutions. hBOA with DHC was applied to systems of size up to $n = 20 \times 20 = 400$.

3.2 Results

Figure 1(a) shows the average number of evaluations for both hBOA alone and hBOA with DHC. The number of evaluations is averaged over all runs for problem instances of the same size. Note that each data point in the plot corresponds to $8 \times 30 = 240$ runs. Additionally, the figure shows



(a) Two dimensions.

(b) Three dimensions

Figure 1: hBOA on 2D and 3D Ising spin glasses.

the best-fit polynomial approximating the growth of the number of evaluations until the optimum has been found. The total number of evaluations appears to grow polynomially as $O(n^{2.25})$ for hBOA, whereas it is $O(n^{2.73})$ for hBOA with DHC. Despite that the asymptotic complexity grows a little faster for hBOA with DHC, the population sizes and the running times decrease significantly (approximately tenfold) with DHC—for instance, hBOA with DHC can solve a 400-bit problem in *less* evaluations than hBOA needs to solve only a 225-bit problem.

3.3 Discussion

The results presented in Figure 1(a) indicate that hBOA is capable of solving Ising spin-glasses in low order polynomial time. This is good news, but how does the performance of hBOA compare to that of other algorithms for solving the same subclass of spin glasses?

To answer the above question, let us first compute the overall computational complexity of hBOA without DHC. Each evaluation of a spin-glass state can be bounded by $O(n)$ trivial operations, so the overall time spent in fitness evaluation grows as $O(n^{3.25})$. However, in this case the computational complexity of model building dominates other factors. Using an asymptotic complexity bound computed by Pelikan et al. (Pelikan, Goldberg, & Cantú-Paz, 2000) and the empirical results, the overall time complexity of hBOA can be bounded by $O(n^{4.25})$. If the model was updated incrementally in each generation, the time complexity can be expected to decrease to somewhere between $O(n^{3.25})$ and $O(n^{4.25})$. The complexity of hBOA with DHC can be computed analogically to the case with hBOA alone, yielding a conservative bound of $O(4.73)$.

There are several problem-specific algorithms that attempt to solve the above special case of two-dimensional spin glasses (e.g., Kardar and Saul (1994, De Simone, Diehl, Jünger, and Reinelt (1996, Galluccio and Loebl (1999a, Galluccio and Loebl (1999b)). Most recently, Galluccio and Loebl (Galluccio & Loebl, 1999a; Galluccio & Loebl, 1999b) proposed an algorithm for solving spin glasses in $O(n^{3.5})$ for all graphs with bounded genus (two-dimensional toroids are a special case of graphs with bounded genus). So, the overall time complexity of the best currently known algorithm for the considered class of spin glasses is $O(n^{3.5})$.

The above results indicate that the asymptotic complexity of hBOA is slightly worse than that of the best problem-specific algorithm; in particular, hBOA requires $O(n^{4.25})$ steps without DHC and $O(n^{4.73})$ steps with DHC, whereas the algorithm of Galluccio and Loebl requires only $O(n^{3.5})$ steps. However, hBOA does not use any problem-specific knowledge except for the evaluation of possible states of the system, whereas the method of Galluccio and Loebl fully relies on the knowledge of the problem structure and its properties. Even without requiring any problem-specific information in advance, hBOA is capable of competing with the state-of-the-art methods in the field. Using DHC leads to a speed up that results in running times that are *better* than those reported by Galluccio and Loebl. For instance, using hBOA with DHC to solve 400-bit instances took on average 9.7 minutes per instance on a Pentium II/400MHz (the worst case took about 21 minutes, the best case about 2.92 minutes), while Galluccio and Loebl reported times of about 25 minutes on an Athlon/500MHz. Therefore, hBOA+DHC is capable of finding the optimum significantly faster despite that hBOA+DHC does not assume any particular structure of the problem. Nonetheless, it can be expected that the situation will change for larger systems and the algorithm of Galluccio and Loebl will become superior.

Another important point in favor of hBOA is that hBOA does not explicitly restrict the interaction structure of a problem; consequently, hBOA is applicable to spin glasses in more than two dimensions and other spin glasses that fall outside the scope of the method of Galluccio and Loebl.

3.4 From 2D to 3D

Despite that competent methods exist for solving two-dimensional spin glasses, none of these methods is directly applicable in the three-dimensional case. In fact, finding a ground state of three-dimensional spin glasses is NP-complete even for coupling constants restricted to $\{-1, 0, +1\}$ (Barahona, 1982). Since in our case zero coupling constants are not allowed, instances studied here might be solvable in polynomial time although there is no algorithm that is proven to do that. Nonetheless, since hBOA does not explicitly use the dimensionality of the underlying spin-glass problem, it is straightforward to apply hBOA+DHC to three-dimensional spin glasses.

To test the scalability of hBOA with DHC, eight random spin-glass systems on a three-dimensional cube with periodic boundary conditions were generated for systems of size from $n = 4 \times 4 \times 4 = 64$ to $n = 7 \times 7 \times 7 = 343$ spins. Since no other method exists to verify whether the found state actually represents the ground state, hBOA with DHC was first run on each instance with an extremely large population of orders of magnitude larger than the expected one. After a number of generations, the best solution found was assumed to represent the ground state.

Figure 1(b) shows the number of evaluations until hBOA with DHC found the ground state of the tested three-dimensional Ising spin-glass instances. The overall number of evaluations appears to grow polynomially as $O(n^{3.65})$. That means that increasing the dimensionality of spin-glass systems increases the complexity of solving these systems. However, efficient performance is retained even in three dimensions.

4 MAXSAT

The task of find an interpretation of predicates that maximizes the number of satisfied clauses of a given predicate logic formula expressed in conjunctive normal form (MAXSAT) is an important problem of complexity theory and artificial intelligence. Since MAXSAT is NP-complete in its general form, there is no known algorithm that can solve MAXSAT in worst-case polynomial time.

In the context of GAs MAXSAT is usually used as an example class of problems that *cannot* be efficiently solved using selectorecombinative search (Rana & Whitley, 1998), although some positive results were reported with adaptive fitness (Gottlieb, Marchiori, & Rossi, 2002). The reason for poor GA performance appears to be that short-order partial solutions lead away from the optimum (sometimes in as many as 30% of predicate variables) as hypothesized by Rana and Whitley (Rana & Whitley, 1998). We know that hBOA outperforms traditional GAs on challenging hierarchical problems; will hBOA do better in the MAXSAT domain as well?

Here we consider the case where each formula is given in conjunctive normal form with clauses of length at most k; formulas in this form are called k-CNFs. A CNF formula is a *logical and* of clauses, where each clause is a *logical or* of k or less literals. Each literal is either a predicate or a negation of a predicate. An example 3-CNF formula over predicates (variables) X_1 to X_5 is $(X_5 \vee X_1 \vee \neg X_3) \wedge (X_2 \vee X_1) \wedge (\neg X_4 \vee X_1 \vee X_5)$.

An interpretation of predicates assigns each predicate either true or false; for example, $(X_1 = \text{true}, X_2 = \text{true}, X_3 = \text{false}, X_4 = \text{false}, X_4 = \text{true})$ is an interpretation of X_1 to X_5. The task in MAXSAT is to find an interpretation that maximizes the number of satisfied clauses in the given formula. For example, the assignment $(X_1 = \text{true}, X_2 = \text{true}, X_3 = \text{true}, X_4 = \text{true}, X_5 = \text{true})$ satisfies all the clauses in the above formula, and is therefore one of the optima of the corresponding MAXSAT problem. MAXSAT is NP complete for k-CNF if $k \geq 2$.

4.1 Methodology

In hBOA each candidate solution represents an interpretation of predicates in the problem. Each bit in a solution string corresponds to one predicate; true is represented by 1, false is represented by 0. The fitness of a solution is equal to the number of satisfied clauses given the interpretation encoded by the solution. Similarly as earlier, DHC is incorporated into hBOA to improve its performance. DHC for MAXSAT is often called GSAT in the machine learning community (Selman, Levesque, & Mitchell, 1992). All other parameters except for the population size are the same as for spin glasses (see Section 3.1). For each problem instance, a minimal population size required for reliable convergence to the optimum in 30 independent runs was used.

hBOA with DHC is compared to two methods for solving MAXSAT: (1) GSAT, and (2) WalkSAT. GSAT (Selman, Levesque, & Mitchell, 1992) starts with a random solution. In each iteration, GSAT applies a 1-bit flip that improves the current solution the most until no more improvement is possible. WalkSAT extends GSAT to incorporate random changes. In each iteration, WalkSAT performs the greedy step of GSAT with the probability p; otherwise, one of the predicates that are included in some unsatisfied clause is randomly selected and its interpretation is changed. Best results have been obtained with $p = 0.5$, although the optimal choice of p changes from application to application.

4.2 Tested instances

Two types of MAXSAT instances are tested: (1) random satisfiable 3-CNF formulas, and (2) instances of combined-graph coloring translated into MAXSAT. All tested instances have been downloaded from the Satisfiability Library SATLIB (http://www.satlib.org/).

Instances of the first type are randomly generated satisfiable 3-CNF formulas. All instances belong to the phase transition region (Cheeseman, Kanefsky, & Taylor, 1991), where the number of clauses is equal to $4.3n$ (n is the number of predicates). *Random* problems in the phase transition are known to be the most difficult ones for most MAXSAT heuristics (Cheeseman, Kanefsky, & Taylor,

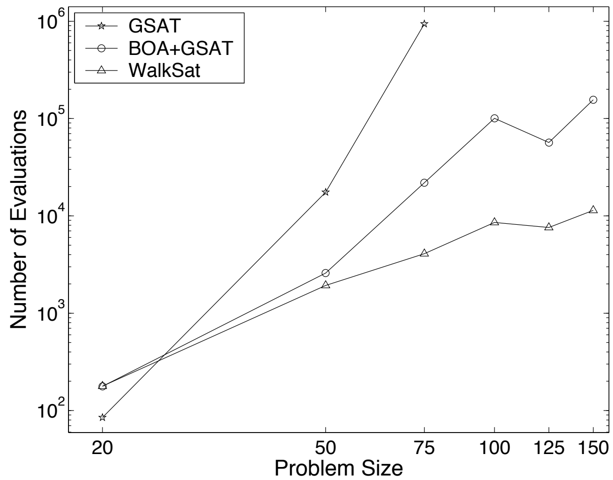


Figure 2: hBOA+GSAT, GSAT, and WalkSAT on random 3-CNF from phase transition.

1991). Satisfiability of randomly generated formulas is not forced by restricting the generation procedure itself, but a complete algorithm for verifying satisfiability such as Satz (Li & Anbulagan, 1997) is used to filter out unsatisfied instances. This results in generation of more difficult problems.

Instances of the second type were generated by translating graph-coloring instances to MAXSAT. In graph coloring, the task is to color the vertices of a given graph using a specified number of colors so that no connected vertices share the same color. Every graph-coloring instance can be mapped into a MAXSAT instance by introducing one predicate for each pair (color, vertex), and creating a formula that is satisfiable if and only if exactly one color is chosen for each vertex, and the colors of the vertices corresponding to each edge are different.

Here, graph-coloring instances are generated by combining regular ring lattices and random graphs with a fixed number of neighbors (Gent, Hoos, Prosser, & Walsh, 1999). Combining two graphs consists of selecting (1) all edges that overlap in the two graphs, (2) a random fraction $(1-p)$ of the remaining edges from the first graph, and (3) a random fraction p of the remaining edges from the second graph. By combining regular graphs with random ones, the amount of structure in the resulting graph can be controlled; the smaller the p, the more regular the graphs are (for $p = 0$, the resulting graph is a regular ring lattice).

For small values of p (from about 0.003 to 0.03), MAXSAT instances of the second type are extremely difficult for WalkSAT and other methods based on local search. Here all instances are created from graphs of 100 vertices and 400 edges that are colorable using 5 colors, and each coloring is encoded using 500 binary variables (predicates).

4.3 Results on random 3-CNFs

Figure 2 compares the performance of hBOA with GSAT, GSAT alone, and WalkSAT. Ten instances are tested for each problem size. More specifically, the first ten instances from the archives downloaded from SATLIB are used.

How does the performance of hBOA+GSAT compare to that of GSAT alone and WalkSAT? GSAT is capable of solving only the simplest instances of up to $n = 75$ variables, because the computational time requirements of GSAT grow extremely fast. Already for instances of $n = 100$ variables, GSAT could not find the optimal interpretation even after days of computation. This leads us to a conclusion that GSAT alone cannot solve the problem efficiently, although it improves

Instance	p	hBOA Evals.	Instance	p	hBOA Evals.
SW100-8-5/sw100-1.cnf	2^{-5}	1,262,018	SW100-8-7/sw100-2.cnf	2^{-7}	1,558,891
SW100-8-5/sw100-2.cnf	2^{-5}	1,099,761	SW100-8-7/sw100-6.cnf	2^{-7}	1,966,648
SW100-8-5/sw100-3.cnf	2^{-5}	1,123,012	SW100-8-7/sw100-7.cnf	2^{-7}	1,222,615
SW100-8-6/sw100-1.cnf	2^{-6}	1,183,518	SW100-8-8/sw100-1.cnf	2^{-8}	1,219,675
SW100-8-6/sw100-2.cnf	2^{-6}	1,324,857	SW100-8-8/sw100-2.cnf	2^{-8}	1,537,094
SW100-8-6/sw100-3.cnf	2^{-6}	1,629,295	SW100-8-8/sw100-6.cnf	2^{-8}	1,650,568
SW100-8-7/sw100-1.cnf	2^{-7}	1,732,697	SW100-8-8/sw100-7.cnf	2^{-8}	1,287,180

Table 1: hBOA with GSAT on WalkSAT-hard MAXSAT instances. WalkSAT could not solve any of these instances even with more than 40,000,000 evaluations.

the efficiency of hBOA when used in the hybrid hBOA+GSAT. The results also indicate that the performance of WalkSAT is slightly better than that of hBOA+GSAT, although performance of the two approaches is comparable. Thus, both selectorecombinative search and randomized local search can tackle random 3-CNFs quite efficiently.

4.4 Results on graph coloring MAXSAT

We've seen that randomly generated 3-CNF instances are rather easy for both selectorecombinative algorithms and local search. Nonetheless, real-world problems are not random, most real-world problems contain a considerable amount of regularities. Combined-graph coloring described in Section 4.2 provides an interesting class of problems where regularity is combined with randomness. By controlling the relative amounts of structure and randomness, interesting classes of problems can be generated. This section tests the algorithms that performed relatively well on random 3-CNF, and applies these algorithms to combined-graph coloring translated into MAXSAT.

Although regular ring lattices ($p = 0$) can be solved by WalkSAT efficiently (Gent, Hoos, Prosser, & Walsh, 1999), introducing even a slight perturbation to the regular graph by combining it with a random graph severely affects WalkSAT's performance. More specifically, WalkSAT is practically unable to solve any instances with $p \leq 2^{-5}$ even with very large number of restarts and trials. It's no surprise that for these problem instances GSAT is also intractable. On the other hand, hBOA+GSAT is capable of solving all these instances despite their large size (500 variables). Table 1 shows the performance of hBOA on several instances that are practically unsolvable by WalkSAT. WalkSAT is not able to solve any of these instances even when allowed to check over 40 million interpretations!

4.5 Discussion

There are several important observations regarding the performance of the hybrid hBOA+GSAT on the tested MAXSAT instances.

hBOA+GSAT outperformed GSAT alone on all problem instances; not surprisingly, hBOA is capable of supplying much better starting points for GSAT than random restarts do. However, on the class of randomly generated 3-CNF, the hybrid hBOA+GSAT is outperformed by a randomized GSAT called WalkSAT. Nonetheless, the performance of hBOA+GSAT is competitive with that of WalkSAT.

On the other hand, for those problem instances that are practically unsolvable using local search (WalkSAT and GSAT), hBOA+GSAT retains efficient performance. In particular, MAXSAT instances obtained by translating graph coloring of graphs with a large amount of structure and a little amount of randomness cannot be solved by GSAT or WalkSAT even after tens of millions of evaluations, whereas hBOA+GSAT is capable of solving all these problems in fewer than two million evaluations. Therefore, hBOA+GSAT can solve those instances that are easy for local search (random 3-CNF), but it is not limited to those instances—it can solve also problems that are practically unsolvable by local search alone.

To summarize, hBOA is able to provide robust solution to two different classes of MAXSAT instances—instances that are completely random and instances that contain a significant amount of regularities and little randomness. This result is made even more important by the fact that hBOA is not given any problem specific knowledge in advance and learns how to solve the problem automatically using only the evaluation procedure.

5 Summary and conclusions

The above results confirmed the assumption that decomposition and hierarchical decomposition is an inherent part of many real-world problems and that effective discovery and exploitation of single-level and hierarchical decomposition should provide for robust and scalable solution of a broad range of optimization problems. hBOA was capable of solving randomly generated Ising spin glass problem instances and two types of MAXSAT problems with competitive or better performance than problem specific approaches.

hBOA was told nothing about the semantics of the problem; initially it didn't know whether it was trying to solve a spin glass, MAXSAT, or any other problem. All problem-specific knowledge was acquired automatically without any interaction with the user. Recently, hBOA was successfully applied to other classes of problems also without any knowledge of the semantics of the problem; these problems included onemax, composed traps, exponentially scaled deceptive problems, and hierarchical traps. Despite of the lack of prior problem-specific knowledge, hBOA was capable of automatic discovery and exploitation of problem regularities that was effective enough to solve the broad range of challenging problems in a robust and scalable manner. This adds a new piece of evidence that hBOA is indeed a robust and scalable optimization technique that should certainly make a difference in current computational optimization practice.

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0163. 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

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

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

Cheeseman, P., Kanefsky, B., & Taylor, W. M. (1991). Where the really hard problems are. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-91)*, 331–337.

De Simone, C., Diehl, M., Jünger, M., & Reinelt, G. (1996). Exact ground states of two-dimensional +−J Ising spin glasses. *Journal of Statistical Physics*, *84*, 1363–1371.

Galluccio, A., & Loebl, M. (1999a). A theory of Pfaffian orientations. I. Perfect matchings and permanents. *Electronic Journal of Combinatorics*, *6*(1). Research Paper 6.

Galluccio, A., & Loebl, M. (1999b). A theory of Pfaffian orientations. II. T-joins, k-cuts, and duality of enumeration. *Electronic Journal of Combinatorics*, *6*(1). Research Paper 7.

Gent, I., Hoos, H. H., Prosser, P., & Walsh, T. (1999). Morphing: Combining structure and randomness. *Proceedings of the American Association of Artificial Intelligence (AAAI-99)*, 654–660.

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.

Gottlieb, J., Marchiori, E., & Rossi, C. (2002). Evolutionary algorithms for the satisfiability problem. *Evolutionary Computation*, *10*(1), 35–50.

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

Harik, G. R. (1994). *Finding multiple solutions in problems of bounded difficulty* (IlliGAL Report No. 94002). Urbana, IL: University of Illinois at Urbana-Champaign.

Kardar, M., & Saul, L. (1994). The 2D +/−J Ising spin glass: Exact partition functions in polynomial time. *Nucl. Phys. B*, *432*, 641–667.

Larranaga, P., & Lozano, J. A. (Eds.) (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Boston, MA: Kluwer.

Li, C. M., & Anbulagan (1997). Heuristics based on unit propagation for satisfiability problems. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-97)*, 366–371.

Monien, B., & Sudborough, I. H. (1988). Min cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, *58*(1–3), 209–229.

Mühlenbein, H., Mahnig, T., & Rodriguez, A. O. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, *5*, 215–247.

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

Naudts, B., & Naudts, J. (1998). The effect of spin-flip symmetry on the performance of the simple GA. *Parallel Problem Solving from Nature*, 67–76.

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

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

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000). Linkage problem, distribution estimation, and Bayesian networks. *Evolutionary Computation*, *8*(3), 311–341. Also IlliGAL Report No. 98013.

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

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

Pelikan, M., & Mühlenbein, H. (1999). The bivariate marginal distribution algorithm. *Advances in Soft Computing—Engineering Design and Manufacturing*, 521–535.

Rana, S., & Whitley, D. L. (1998). Genetic algorithm behavior in the MAXSAT domain. *Parallel Problem Solving from Nature*, 785–794.

Selman, B., Levesque, H. J., & Mitchell, D. (1992). A new method for solving hard satisfiability problems. *Proceedings of the National Conference on Artificial Intelligence (AAAI-92)*, 440–446.

Simon, H. A. (1968). *The sciences of the artificial.* Cambridge, MA: The MIT Press.

Van Hoyweghen, C. (2001). Detecting spin-flip symmetry in optimization problems. In Kallel, L., et al. (Eds.), *Theoretical Aspects of Evolutionary Computing* (pp. 423–437). Berlin: Springer.