Missouri Estimation of Distribution Algorithms Laboratory

## Effects of a Deterministic Hill climber on hBOA

Elizabeth Radetic, Martin Pelikan, and David E. Goldberg

MEDAL Report No. 2009003

January 2009


### Abstract

Hybridization of global and local search algorithms is a well-established technique for enhancing the efficiency of search algorithms. Hybridizing estimation of distribution algorithms (EDAs) has been repeatedly shown to produce better performance than either the global or local search algorithm alone. The hierarchical Bayesian optimization algorithm (hBOA) is an advanced EDA which has previously been shown to benefit from hybridization with a local searcher. This paper examines the effects of combining hBOA with a deterministic hill climber (DHC). Experiments reveal that allowing DHC to find the local optima makes model building and decision making much easier for hBOA. This reduces the minimum population size required to find the global optimum, which substantially improves overall performance.



### Keywords

Hierarchical BOA, local search, spin glass, trap-5, MAXSAT, hybrid evolutionary algorithms, estimation of distribution algorithms, efficiency enhancement.


Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Department of Mathematics and Computer Science
University of Missouri–St. Louis
One University Blvd., St. Louis, MO 63121
E-mail: medal@cs.umsl.edu
WWW: http://medal.cs.umsl.edu/

# Effects of a Deterministic Hill climber on hBOA

**Elizabeth Radetic**
Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math and Computer Science, 321 CCB
University of Missouri at St. Louis
One University Blvd., St. Louis, MO 63121
err26c@umsl.edu

**Martin Pelikan**
Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math and Computer Science, 320 CCB
University of Missouri at St. Louis
One University Blvd., St. Louis, MO 63121
pelikan@cs.umsl.edu

**David E. Goldberg**
Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign, Urbana, IL 61801
deg@uiuc.edu

**Abstract**

Hybridization of global and local search algorithms is a well-established technique for enhancing the efficiency of search algorithms. Hybridizing estimation of distribution algorithms (EDAs) has been repeatedly shown to produce better performance than either the global or local search algorithm alone. The hierarchical Bayesian optimization algorithm (hBOA) is an advanced EDA which has previously been shown to benefit from hybridization with a local searcher. This paper examines the effects of combining hBOA with a deterministic hill climber (DHC). Experiments reveal that allowing DHC to find the local optima makes model building and decision making much easier for hBOA. This reduces the minimum population size required to find the global optimum, which substantially improves overall performance.

**Keywords:** Hierarchical BOA, local search, spin glass, trap-5, MAXSAT, hybrid evolutionary algorithms, estimation of distribution algorithms, efficiency enhancement.

## 1 Introduction

It is common practice to incorporate some form of local search into evolutionary algorithms in order to improve performance (Moscato, 1989; Davis, 1991; Ibaraki, 1997; Goldberg & Voessner, 1999; Sinha & Goldberg, 2003). Theoretical models have been created to express the optimal balance between global and local search (Goldberg & Voessner, 1999; Sinha & Goldberg, 2001). These models analyze the relationship of global and local search in terms of time steps required to achieve solutions of a specified accuracy. Nevertheless, theoretical models are often difficult to apply in practice. Thus empirical studies are also required, both to verify the theory and to provide practical guidance to practitioners.

The purpose of this paper is to present an in-depth empirical analysis of the effects of incorporating a deterministic hill climber (DHC) into the hierarchical Bayesian optimization algorithm (hBOA). The results confirm that incorporating DHC into hBOA substantially improves its performance.

This paper is organized as follows. Section 2 explains the background of hybrid evolutionary algorithms and motivates this study. Section 3 describes the methodology of the experiments done in this study, including the algorithms and their parameters. Section 4 summarizes the test problems used. Section 5 presents the results of the experiments and section 6 then provides a discussion of these results. Section 7 concludes the paper by reviewing the results presented and highlighting their importance.

## 2 Background & Motivation

Hybridization has, justifiably, been a subject of great interest in a variety of studies into optimization. Global-local search hybrids combine a global searcher, such as a genetic algorithm (GA) or estimation of distribution algorithm (EDA), with a local search algorithm, such as a hill climber, to increase increase efficiency over either algorithm in isolation (Moscato, 1989; Davis, 1991; Ibaraki, 1997; Goldberg & Voessner, 1999; Sinha & Goldberg, 2003).

Global-local hybrid algorithms accomplish this by exploiting the separate strengths of both local and global searchers. Global searchers are proficient at providing a comprehensive overview of the fitness landscape to find many different basins of attraction. Local searchers, on the other hand, have proven quite effective at quickly zeroing in on local optima once the basins of attraction have been found (Sinha & Goldberg, 2003). Thus, local search allows the global searcher to focus its search on basins of attraction rather than on the entire fitness landscape.

While many combinations of global and local searchers have proven more effective than the individual searchers alone, there are some trade offs to consider when combining them. Caution must be exercised so as not to waste too much time in local search that does not substantially assist the global search; for example, the local searcher should not spend too much time in the same basin of attraction (Sinha & Goldberg, 2003). Another effect of local search is the reduction in variance, which can have both positive and negative consequences. On the one hand, variance reduction reduces collateral noise, which in turn makes decision making easier (Goldberg, 2002). For EDAs, variance reduction also improves model building by strengthening the statistical dependencies between related bits. These improvements come at a price, however, as population diversity may be lost too quickly, potentially restricting the search to the basins of attraction which have already been discovered. This might prevent the global optimum from ever being reached. It is therefore important to find a balance which will allow us to exploit the benefits of both the local and global search to achieve the optimum overall performance while minimizing negative consequences.

Many studies have been conducted to explore the effects of hybridization and to guide practitioners in its application. Goldberg and Voessner (1999) and Sinha and Goldberg (2001) laid some theoretical groundwork by analyzing global-local search hybrids in terms of the number of local versus global search steps. This analysis resulted in models which suggest a way to find the optimal balance between the two. Later, Sinha and Goldberg (2003) provided a comprehensive survey of the field of hybrid genetic and evolutionary algorithms (GEAs), categorizing and analyzing them according to such features as their architecture and purpose, and secondary search method. The survey then presents some of the design issues associated with using hybrid GEAs in practice. Of particular note is the issue of the effects of local search on sampling in GEAs, and the question of how long to allow local search to operate.

A common thread throughout these studies is that combining local and global searchers can produce better results than either of the two methods alone is capable of achieving. To enforce this point, Sinha and Goldberg (2003) highlight several studies of interest. One study by Hart (1994) combined local search with GAs and described various methods for choosing members of the population to use in local search and the duration to allow local search to run in order to achieve the best performance. This work found that adaptive methods for selecting participants for local search generally outperformed methods using a fixed percentage of the population, including the typical practice of selecting 100% of individuals for local search. The issue of duration of local search was also examined; however, in the results presented, local search frequency seemed to outweigh duration in each of the test functions.

Land (1998) extended this work by comparing various local search methods both in isolation and as part of a hybrid GA. The study examined the results of varying the parameters of each algorithm, including the ratio of local to global search, the number of local search steps to perform, and the methods for selecting population members for local search. Results revealed that running the local search longer allowed good solutions to be found faster, but that the overall ratio of local to global search had a greater impact on final solution quality. They also demonstrated that the different methods presented for selecting solutions for local search had little impact on the final solution quality.

In addition to GAs, a great deal of work has also been done in hybridizing EDAs. Many different EDAs have been successfully combined with various local search techniques to solve a variety of problems. These include cluster optimization (Sastry, 2001a), spin glasses and MAXSAT (Pelikan, 2002; Pelikan & Goldberg, 2003), graph bipartitioning (Mühlenbein & Mahnig, 2002), optimization of non-convex functions (Zhang, Sun, Tsang, & Ford, 2004), quadratic assignment (Tsutsui, Pelikan, & Ghosh, 2006; Zhang, Sun, Tsang, & Ford, 2006), the traveling salesman problem (Tsutsui, Pelikan, & Ghosh, 2006), and nurse rostering (Aickelin, Burke, & Li, 2007). In each case, the local searcher provided a significant enhancement of efficiency for the EDA, once again demonstrating the benefits of global-local search hybridization.

The following experiments presented in this paper were conducted using the hierarchical Bayesian optimization algorithm (hBOA) (Pelikan, 2005). hBOA is an advanced EDA which has been shown to be an efficient black-box optimizer on a variety of difficult problems (Pelikan & Goldberg, 2001; Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006; Pelikan, Katzgraber, & Kobe, 2008).

hBOA has been frequently used in combination with local search and its performance has been studied on a variety of problems including Ising spin glasses and MAXSAT (Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006). In the case of Ising spin glasses, using hBOA with a discrete hill climber has been shown to reduce the population size, running time and number of evaluations and to permit efficient evaluation of much larger problems than with hBOA alone (Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006). To study MAXSAT, hBOA was again combined with the hill climber and this hybrid was shown to solve problem instances unsolvable by the local searcher alone (Pelikan & Goldberg, 2003). This and other studies (Pelikan, Katzgraber, & Kobe, 2008) reaffirm the value provided by the local search in terms of the improvements in performance which it facilitates.

The aforementioned research demonstrates that hybridizing hBOA and other EDAs has proven to be quite beneficial. There are, however, several outstanding questions which still need to be addressed. One issue is the question of how frequently to apply local search, and how long to let it run. Some studies have addressed this for GAs and EDAs with various local searchers (Hart, 1994; Land, 1998; Lima, Pelikan, Sastry, Butz, & Goldberg, 2006). The results, however, did not provide a consistent prescription for the optimal settings of global-local search hybrids of EDAs in the general case.

Additional issues to consider concern the positive and negative effects of the variance reduction provided by local search. As noted above, this reduction can have beneficial consequences for decision making and model building; however, it also has the potential to impair the search for the global optimum. It is therefore important to explore which of these effects exerts a more dominant influence on the performance of the hBOA-DHC hybrid.

This paper addresses these issues by examining the effects of combining hBOA with a deterministic hill climber. Deterministic hill climbing is a simple local search technique, which can be easily incorporated into hBOA for most problems, yet the benefits it provides span across a broad range of difficult problem classes. In the experiments conducted, both the frequency and duration of the local search are varied. In addition, the effects of unrestricted DHC are examined. The results will provide some insight into the role DHC plays in improving hBOA's performance.

The following sections describe the algorithms used in this study (section 3), as well as the different problems tested (section 4). Subsequent sections (sections 5 and 6) describe and discuss the results of the experiments.

# 3 Algorithms

This section describes the methodology of this empirical study. Section 3.1 describes hBOA, the global search method employed, and how its parameters were set. Section 3.2 describes the local searcher, the deterministic hill climber.

## 3.1 Hierarchical BOA (hBOA)

As previously stated, the hierarchical Bayesian optimization algorithm (hBOA) was the global search method used in these experiments. Hierarchical BOA (Pelikan & Goldberg, 2001; Pelikan, 2005) starts with a randomly-generated population of binary strings. It uses a Bayesian network with decision graphs to build a probabilistic model of promising solutions selected from the population, then samples the network to generate new ones. These offspring are then incorporated into the original population using restricted tournament replacement (Harik, 1995). This algorithm has previously proven to be very effective at solving a variety of difficult problems (Pelikan & Goldberg, 2001; Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006), even more so when combined with the deterministic hill climber.

The experiments described here examined the effects of combining hBOA with a deterministic hill climber (DHC), with varying degrees of restriction. hBOA's parameters were set as follows. Truncation selection was used to select the top 50% of the population to use as parents for the next generation. The maximum number of generations was limited to 300, although the vast majority of cases required only a fraction of this number. The algorithm was run until a solution of sufficient quality was found, with the specific criteria varying by problem, as described below. The bisection method (Sastry, 2001b; Pelikan, 2005) was used to estimate an optimal population size for each problem instance and for each set of local search parameters.

### 3.2 Deterministic Hill Climber (DHC)

For this study, a deterministic hill climber (DHC) was combined with hBOA to investigate the effects of local search on hBOA's performance. Each step of the deterministic hill climber flips the bit in the string that will produce the maximum improvement in fitness value. This process can be allowed to iterate until no single bit flip produces additional improvement, or it can be stopped earlier, such as after a set number of flips or when a specified solution quality has been obtained. Although it may not be the most advanced method of local search, DHC is a robust method which can be used on any problem, without requiring problem-specific knowledge. It can therefore be a useful component of hybrid black-box optimization algorithms.

These experiments focused on the effects of varying the frequency and duration of DHC, as well as the impact of restricted and unrestricted DHC on hBOA's performance. The hill climbing procedure was performed on each evaluated string with a probability $p_{dhc}$ from 0% to 100% in increments of 10%. The maximum number of bits to flip on a string was also restricted, from 0 to the number of bits in the string, in increments of 10% of the total length of the string. Results produced with each set of parameters were then compared to discover which led to the most improvement.

The next section will outline the test problems.

## 4 Test Problems

This section will describe each test problem used in this study. The problems are the order-5 trap (section 4.1), the Ising spin glass (section 4.2), and MAXSAT (section 4.3).

### 4.1 Trap-5

The first problem tested was the concatenated trap of order 5 (trap-5) (Ackley, 1987; Deb & Goldberg, 1994). Trap-5 divides a string into non-overlapping subproblems of 5 consecutive bits. The fitness contribution of each of these groups of bits is determined as follows:

$$f_{trap_5}(u) = \begin{cases} 5 & \text{if } u = 5, \\ 4 - u & \text{otherwise} \end{cases}, \tag{1}$$

where $u$ is the number of ones in the subproblem. This function is summed over all subproblems to determine the fitness of the entire string.

Traps serve as excellent test problems for EDAs because they consist of independent subproblems of bounded order. These subproblems are fully deceptive (Deb & Goldberg, 1994; Goldberg, 2002) and that is why their bits must be treated together as a block. Any algorithm which cannot discover higher-order building blocks but considers only the individual bits, such as DHC or a GA using uniform crossover, would exhibit extremely poor performance on this problem (Thierens & Goldberg, 1993; Harik & Goldberg, 1996; Thierens, 1999; Goldberg, 2002). Thus trap-5 provides a straightforward test of an algorithm's ability to decompose the problem correctly. In order to solve this problem to optimality, an EDA must be able to discover and maintain the relationships between different bits in the same building block while effectively recombining unrelated blocks (Goldberg, 2002; Pelikan, 2005). hBOA is abundantly capable of solving trap-5 to optimality; however, DHC, on its own, generally would not be able to find the global optimum. It will therefore be interesting to see whether DHC improves the performance of hBOA on this problem.

In these experiments, problems of size 100 to 350 bits were tested, in increments of 50 (20 to 70 5-bit subproblems per string). For each set of parameters, ten independent bisections of ten runs each (100 runs total) were averaged together to minimize noise. A run was considered successful when a string was found consisting of optimal building blocks in all but one trap partition. This relaxation allowed for more stable results. Initially, experiments were performed in which all trap partitions were required to have optimal building blocks; however, this resulted in outliers, and in general results were extremely noisy. Even under these conditions, however, DHC was shown to produce essentially the same effects as in the results presented here.

### 4.2 Ising Spin Glasses

The next problem examined was the 2D Ising spin glass with $\pm J$ couplings and periodic boundary conditions (Binder & Young, 1986; Mezard, Parisi, & Virasoro, 1987; Fischer & Hertz, 1991; Young, 1998). This problem consists of a set of spins, each of which can have a value from $\{+1, -1\}$, and coupling constants relating pairs of spins. The energy, specified as

$$H(\sigma) = -\sum_{i,j=0}^{n} J_{ij}\sigma_i\sigma_j \ , \tag{2}$$

must be minimized. When the overall energy of the system is minimized, the optimum, or ground state, is achieved. Spins are encoded using bit strings, with a 1 representing $+1$, and a 0 representing $-1$.

The Ising spin glass exhibits a complex fitness landscape containing many local optima. Moreover, global optima are often surrounded by areas of very poor fitness. Unlike traps, spin glasses cannot be decomposed into independent subproblems of bounded order (Mühlenbein, Mahnig, & Rodriguez, 1999). This means that any algorithm able to find the global optimum successfully and efficiently must be able to discover the complex interactions among the variables. These properties make it a very difficult problem for local searchers to find ground states on their own, including state-of-the-art Markov chain Monte Carlo methods with Wang-Landau sampling (Dayal, Trebst, Wessel, Würtz, Troyer, Sabhapandit, & Coppersmith, 2004). Yet adding local search such as DHC has proven very effective in enhancing the performance of hBOA on this problem in the past (Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006).

For this study, 5 different sizes of spin glass were examined: $8\times8$ (64 spins), $10\times10$ (100 spins), $12\times12$ (144 spins), $14\times14$ (196 spins) and $16\times16$ (256 spins). For each problem instance, bisection was used to determine the optimal population size. Bisection was terminated upon successful completion of 10 consecutive runs. Results from each of the 10 runs and for 1000 different spin glass instances of each size were averaged together. Instances were obtained from S. Sabhapandit and S. Coppersmith (Dayal, Trebst, Wessel, Würtz, Troyer, Sabhapandit, & Coppersmith, 2004) of the University of Wisconsin, who also identified their ground states. Runs were considered successful when an individual was obtained whose fitness matched the precomputed ground state.

### 4.3 MAXSAT

The task in the maximum satisfiability problem (MAXSAT) is to find an assignment of Boolean variables to satisfy the maximum number of clauses of a formula in conjunctive normal form (CNF). Strings of bits represent the variables in the clauses, with a 0 representing false and a 1 representing true.

The instances in our experiments are comprised of 3-CNF formulas. They consist of conjunctions of clauses composed of disjunctions of 3 literals. MAXSAT instances of this form exhibit a phase transition property; this phase transition occurs when the number of clauses is approximately $4.26n$, or slightly higher for small $n$, where $n$ is the number of distinct variables. For a smaller number of clauses, instances are more easily satisfiable and it is easy to determine that an instance can be satisfied; for a much larger number of clauses, almost all instances are unsatisfiable and it becomes relatively easy to prove this (Hoos & Stützle, 2000). MAXSAT for 3-CNF formulas is NP-complete (Karp, 1972).

MAXSAT is an important problem in theoretical computer science and artificial intelligence which has been studied extensively (Hoos & Stützle, 2000). Like spin glasses, MAXSAT cannot be decomposed into independent subproblems of bounded order. Furthermore, partial solutions tend to lead away from the optimum (Rana & Whitley, 1998), making this a difficult problem for selectorecombinative genetic algorithms. The highly multimodal landscape makes this problem difficult also for most local search techniques, especially for certain classes of formulas. Although hBOA with DHC has been shown to work on this problem in the past (Pelikan & Goldberg, 2003), it will be interesting to see what role DHC plays in efficiently solving MAXSAT instances.

For this project, unforced satisfiable random 3-SAT instances of 50 and 75 variables (218 and 325 clauses, respectively) were used. The number of clauses was determined according to the phase transition values for each problem size, to ensure the instances were as difficult as possible for both systematic SAT solvers and stochastic search algorithms. All instances were downloaded from the Satisfiability Library (SATLIB) (Hoos & Stützle, 2000).

For each instance, bisection was performed to estimate the optimal population size. Bisection was terminated after completion of 10 consecutive successful runs, with results of each run averaged together. 100 randomly-generated, satisfiable instances for each problem size were tested and their results averaged together. Success was achieved in a run when at least one of the ground states was found.

The following section will discuss the results of the experiments performed on each of these problems.

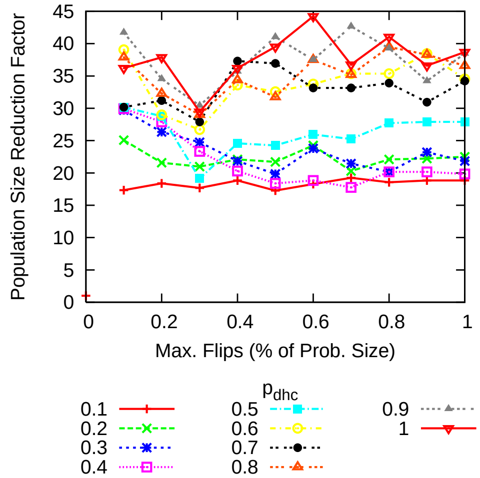
# 5 Results

This section describes the results of the experiments described above. Section 5.1 provides an overview of the results. Sections 5.2 through 5.4 focus on the results for each specific problem.

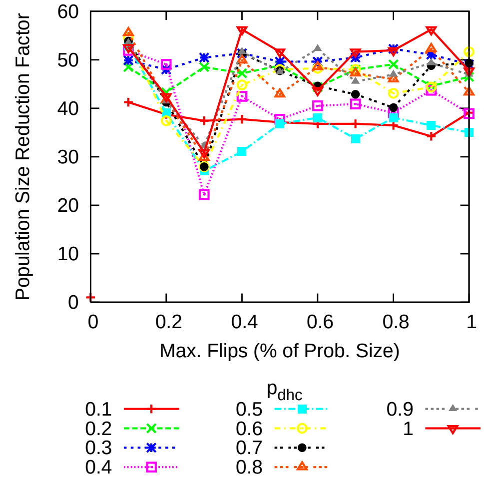
## 5.1 Overview

The following results demonstrate the dramatic improvements obtained by using the deterministic hill climber with hBOA on each of the problems tested.

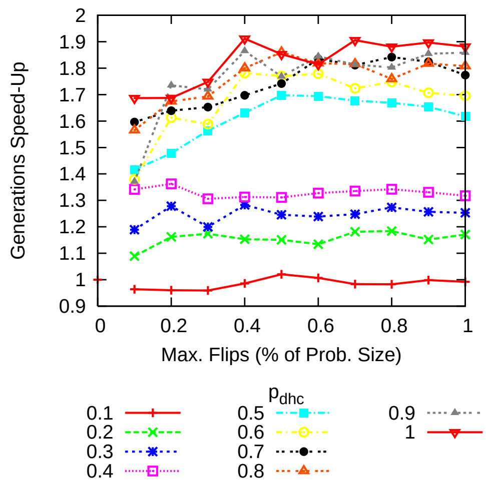
The figures presented illustrate the speedups or reductions obtained for population sizes, numbers of generations, numbers of evaluations and execution times. The speedups and reduction factors represent the ratio of the original values obtained without DHC to those obtained with DHC. For example, if the original population size without DHC is 2000, and the population size with DHC is 200, this indicates a reduction factor of $2000/200 = 10$. Figures compare the outcomes of using DHC on varying proportions of the population and with a varying maximum limit on the numbers of DHC steps (flips). Additional figures compare the results obtained with and without DHC with respect to problem size. The results also focus on the performance of the hBOA-DHC hybrid and the effects of DHC on model building during different stages of the run.



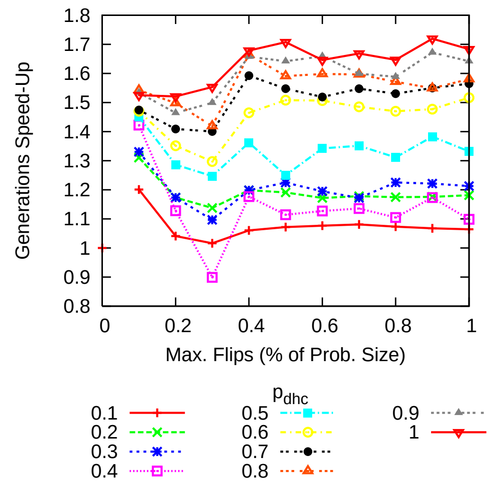

(a) Population size reduction for 100-bit trap-5.




(b) Population size reduction for 350-bit trap-5.



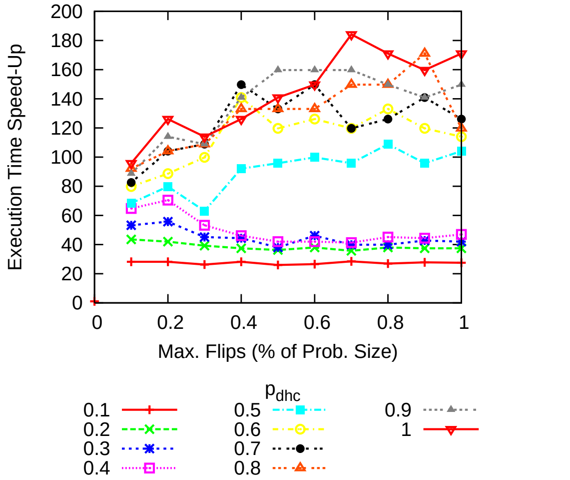

(c) Reduction in the number of generations for 100-bit trap-5.



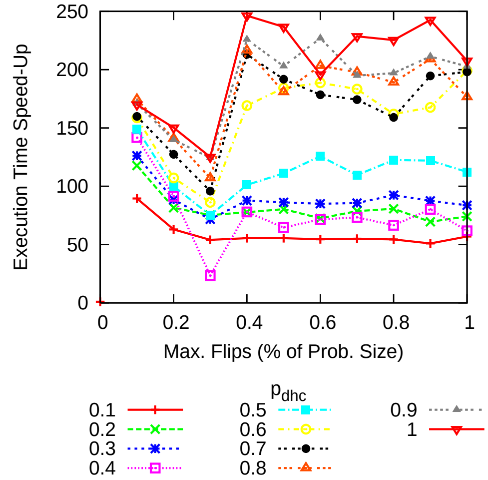

(d) Reduction in the number of generations for 350-bit trap-5.

Figure 1: These graphs show the factor by which the population size and the number of generations were reduced by incorporating DHC into hBOA on trap-5. Results for the smallest and largest problem sizes tested are shown here; improvements were similar for other problem sizes.

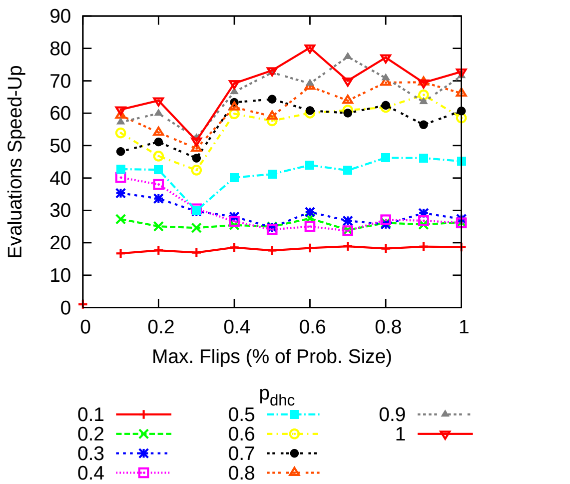
The following sections present in detail the results obtained for using DHC with hBOA when solving trap-5, spin glasses, and MAXSAT. Each section highlights the speedups obtained by combining DHC with hBOA on the problem in question. In each case, results demonstrate the effects of varying the probability of applying DHC as well as the limit on the maximum number of flips allowed. The effects of DHC on different problem sizes of each type are also compared. Finally, each section discusses the number of flips performed in each generation and compares the average number of flips performed to the number of fitness evaluations in each run.



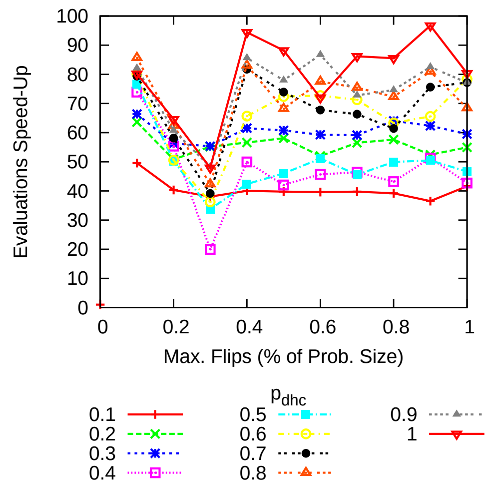

(a) Execution time speedups for 100-bit trap-5.




(b) Execution time speedups for 350-bit trap-5.




(c) Reduction in the number of fitness evaluations for 100-bit trap-5.




(d) Reduction in the number of fitness evaluations for 350-bit trap-5.

Figure 2: Improvements in execution times and the number of evaluations for trap-5 of 100 and 350 bits are shown here. Similar improvements were seen for other problem sizes.

## 5.2 Trap-5

Incorporating the deterministic hill climber into the evaluation of the trap-5 problem produced striking reductions in execution time, number of evaluations, and population size required to reach the specified solution quality. Figure 1 shows reductions in the population size by factors of 15–45, with even greater improvements on the larger problem size (350 bits). On the 100-bit problem, when DHC was run to optimality on the entire population, hBOA required only 1/15 to 1/45 of the population size required without DHC. This translates to speedups of 20 to 180 in execution times and 20 to 80 in the number of evaluations (figure 2). For bigger problems, improvements get even more significant. The number of generations required to achieve the specified solution quality was also reduced. However, these improvements were not nearly as significant as those achieved for the population sizes; figure 1 illustrates a reduction factor of less than 2 for the 100-bit problem. While the gains from the reduction of the population size are more significant for larger problems, the larger (350-bit) problem actually shows a smaller reduction in the number of generations compared to the smaller problem.

It is readily apparent that all runs using DHC to some degree produced a drastic reduction in population size compared to runs which did not use DHC at all, with corresponding drops in both the number of evaluations and execution times. Generally, greater reductions were seen when larger proportions of the population were subjected to the hill climber. In addition, allowing the hill climber to run until the local optimum generally produced at least as much improvement as stopping it earlier. There does, however, seem to be a slight decrease in improvement when only 30% of the bits are flipped, which is most evident for the larger problem sizes.

These improvements are particularly noteworthy since, on its own, the deterministic hill climber would not be expected to find the global optimum for this problem, but rather would push the solution to the deceptive local optimum for most of the trap subproblems. Instead, the hill climber's rush to local optima proves to be an asset here, as it makes the decision making and model building easier, and it simplifies the fitness landscape.

Figure 4 illustrates the average amount of work performed by DHC on a trap of 350 bits; the amount of work done by DHC is measured as the number of flips either per evaluation (part a) or per generation (part b). Figure 4(a) shows the average number of flips per fitness evaluation. As this figure illustrates, when averaged over the entire run, relatively few flips were performed per evaluation. This agrees with figures 1 and 2 in that, regardless of the maximum number flips allowed, relatively few were actually needed before the local optimum was reached. As figure 4(b) shows, when DHC was not restricted, the vast majority of flips were performed in the very first generation. This is a predictable outcome, as DHC was allowed to reach the local optimum, and in subsequent generations most blocks of bits corresponding to trap subproblems would be fixed to either 0s or 1s; consequently, DHC's single flips would not produce any improvement at this point.

### 5.3 Ising Spin Glass

The benefits of using the deterministic hill climber are illustrated clearly in the results of the experiments on spin glasses. Figures 5 and 6 show reductions of the population size by factors of 4-11 and speedups in execution time and numbers of evaluations of approximately 5-35 and 5-50, respectively, on the smaller ($10 \times 10$) problem. As with the previous results, the reduction in the number of generations was much smaller, from 1.5-5 for the $10 \times 10$ spin glass.

Figure 7 shows that, similarly as for trap-5, reductions in population size, as well as in execution times and evaluations, increased with increasing problem size. On the other hand, the improvements in the number of generations decreased slightly as problem sizes increased. This again points population size as the key factor in determining performance gains.

As these results show, while allowing DHC to operate on a greater proportion of the population produced greater speedups, improvements leveled off as greater numbers of flips were allowed. This indicates that, although the local search was permitted to run longer, it found the local optimum after flipping around 30% of the bits in each string. Figure 8 supports this conclusion by illustrating that the amount of work performed by the deterministic hill climber also leveled off at around this same percentage. Moreover, as with the trap, most of the flips were performed in the earliest generations, with a dramatic reduction in the number of flips performed later in the run.

These results indicate that using the deterministic hill climber on 100% of the population and allowing it to run until it found the local optimum produced the greatest improvement. Compared to trap-5, a much clearer correlation can be seen between the extent to which DHC was used and the benefits obtained from it, as is illustrated in Figures 5 and 6. Whereas a more restricted local search might prove beneficial in other situations, these results demonstrate the benefit of allowing the hill climber to operate on the entire population, and to be run until no more improvement is possible.

Although the spin glass problem is quite different from trap-5 and has a much more complex fitness landscape, DHC still produced substantial speedups. Again, these improvements are obtained despite the relatively poor performance of DHC on its own when solving spin glasses. This supports the earlier empirical results which found hBOA with DHC to perform more efficiently than either algorithm alone on this problem (Pelikan & Goldberg, 2003; Pelikan & Hartmann, 2006). Once again, by reducing the fitness landscape to local optima, the hill climber aided hBOA enough to significantly improve the overall performance of the algorithm.

## 5.4 MAXSAT

The deterministic hill climber proved indispensable in the evaluation of the MAXSAT instances. Some of the instances were barely solvable without DHC, and some were even quite challenging when the DHC was used only sparingly. Because of this, experiments were restricted to relatively small problem sizes of no more than 75 variables. Larger problems of up to 250 variables were attempted initially, but most instances could not be solved at all without local search, even with population sizes greater than 800,000. This outcome alone serves as evidence of the value of combining hBOA with the hill climber. Even on the smaller instances presented here, however, the benefits of the hill climber are readily evident.

The improvements gained by using DHC on MAXSAT resemble those seen in the spin glass problem, although they are much greater. Figure 9 shows reductions in the population size of 5-50 for the smaller (50 bit) problem, with the improvements nearly doubling on the larger (75 bit) problem. Similarly, execution times and numbers of evaluations were cut by up to 300 and 200, respectively, on the 50-bit problem. As with the previous problems, although fewer generations of hBOA were also required, these improvements were overshadowed by the much more substantial improvements in population sizes. The number of generations was reduced by no more than 4.5 for the 50-bit problem. The 75-bit problem showed an even smaller reduction in the number of generations, of at most 2.8; for the smaller values of $p_{dhc}$, a few more generations were actually required than when DHC was not used at all on this problem size.

Once again, the average number of flips per evaluation was relatively low (figure 11), which is consistent with the previous problems. Compared to the spin glass and trap problems, a somewhat higher number of flips was maintained in the later generations; however, figure 11(b) clearly shows that a much greater number of flips was performed in the first generation, with a steep decline thereafter.

As with results for the trap and spin glass, the most improvement was gained by submitting every individual in the population to the local search, and allowing the local search to run until the local optimum is found. These results are noisier than those for the spin glasses, however, due to the very small problem sizes tested.

The results for MAXSAT provide yet another example of a global-local search hybrid performing better than either of its constituent algorithms alone. Once again, while DHC by itself is generally incapable of solving MAXSAT efficiently, it provides significant speedups when used in a hybrid with hBOA. What is perhaps even more striking, however, is that this problem was even difficult for hBOA on its own. Although the combination of hBOA with DHC is capable of solving much larger instances of this type (Pelikan & Goldberg, 2003), without local search some instances as small as 100 bits were not solvable using a reasonable population size. These results make an even stronger case for the value of hybridizing selectorecombinative algorithms such as hBOA with simple local searchers such as DHC to boost overall performance.

# 6 Discussion

Results from all three test problems strongly reinforce the value of combining hBOA with the deterministic hill climber. On every size of every problem, hBOA with DHC consistently outperformed hBOA without DHC in every area. Substantial benefits are obtained despite the fact that each one of these problem types is generally difficult for a local searcher such as DHC, which on its own would usually be expected to get stuck in one of the inferior local optima.

Allowing DHC to operate until it reaches a local optimum reduces fitness variance substantially. On one hand, the reduction in variance leads to a loss of population diversity, which in some situations could hamper the search for the global optimum. On the other hand, the reduction in variance makes the decision making (Goldberg & Rudnick, 1991; Harik, Cantú-Paz, Goldberg, & Miller, 1999; Goldberg, 2002) easier because the algorithm only needs to choose between local optima and other, less-fit competitors are eliminated. Even more importantly, reduced variance also makes model building easier for an EDA like hBOA because reduced variance strengthens the statistical dependencies between correlated variables, making them easier for the algorithm to discover and exploit. When these dependencies are easier to discover, the algorithm is able to find them with a smaller population than would otherwise be required. This is significant because model building represents the most important factor influencing the population sizing in hBOA (Pelikan, Goldberg, & Cantú-Paz, 2000; Yu, Sastry, Goldberg, & Pelikan, 2007).

The effects of local search on model building can be illustrated on the trap problem. After the hill climber has been run to local optimality on a solution string, regardless of the starting point, all 5-bit trap partitions of the string will consist of either 00000 or 11111. Because the basin of attraction around 00000 is larger than that around 11111, the majority of flips performed by the hill climber result in an overall loss of 1s in favor of 0s; however, the number of optimal building blocks of all 1s will also be increased. At the same time, less fit competitors will have been eliminated entirely. With strings consisting only of blocks of 0s and 1s, the signal pointing to the discovery of trap partitions becomes extremely clear. Furthermore, to fully encode dependence between all bits in one trap partition under the assumption that each trap partition is set to either 00000 or 11111, much fewer dependencies have to be covered as was already argued elsewhere (Hauschild, Pelikan, Lima, & Sastry, 2007).

To support this hypothesis, the effects of DHC on model building in hBOA are visualized in figures 12 and 13. Figure 12 shows the relative decrease in the proportions of blocks 11111 in different trap partitions, whereas figure 13 shows the number of correct dependencies (those between bits in one trap partition) and the number of spurious dependencies (those between different trap partitions). These results show that, with DHC, hBOA is easily able to find the minimum necessary number of dependencies in the very first generation, even with very small population sizes. In addition, the number of spurious dependencies discovered is extremely small. In contrast, without DHC, very few of the correct dependencies are discovered in the first generation unless the population size is increased substantially. Additionally, many unnecessary dependencies are created for all but very large population sizes. This makes it much more difficult to preserve and juxtapose optimal combinations of bits.

In the absence of DHC, a larger population is needed to provide a sufficient sample to build the model, and additional generations are required so that the necessary variance reduction can take place. After several more generations, the variance and therefore noise have been reduced to a point where the model can be built with a reasonable population size without DHC. Figures 12 and 13 show that this has occurred by approximately the middle of the run. Hauschild, Pelikan, Lima, and Sastry (2007) demonstrated that hBOA discovers most of the necessary dependencies for trap-5 by the middle of the run when the optimal population size is used. Here we see that by reducing variance right away, DHC saves a lot of work that would otherwise require several more generations and a much larger population.

The effects of DHC on decision making can be isolated and analyzed using a model that perfectly fits the structure of trap-5. The perfect model for trap-5 contains dependencies between all 5 bits of each trap partition, but none between bits in different partitions. Thus the 5 bits of each partition are treated as one indivisible block. Figure 14 illustrates the results of solving trap-5 using this ideal model for recombination rather than allowing hBOA to discover the model itself. Even with the perfect model, the benefits of using DHC are substantial. This confirms the hypothesis based on the population sizing theory for selectorecombinative genetic algorithms (Harik et al., 1999), which suggests that the decision making itself becomes easier with DHC. Nonetheless, the reduction in population sizes for the ideal model is substantially smaller than that for standard hBOA, indicating that model building is indeed the most important factor in hBOA population sizing. All of these results clearly demonstrate that the primary benefit of using DHC in hBOA is achieved through the positive effects of DHC on hBOA's model building abilities.

# 7 Summary & Conclusions

This paper has presented a study of the effects of using a deterministic hill climber with hBOA on three different problems: trap-5, 2D Ising spin glasses, and MAXSAT. The paper described the methodology used and the problems tested in this study, including the global and local search algorithms used and the set ups for each problem type. It then presented the experimental results for each of the problems and provided a brief discussion of the outcomes.

All of the problems tested here are challenging for local search methods such as DHC, yet when DHC is used as a component of hBOA, the hybrid demonstrates substantially better performance than either algorithm alone. The results presented indicated unambiguously that, on each of the three problems, using DHC produced significant speedups in every area, including population size, number of evaluations, execution times, and, to a lesser extent, number of generations. Although the frequency and duration of local search were varied in these experiments, for the most part the best results were obtained when DHC was used on 100% of the population, and run until the local optimum was reached.

The benefits of using DHC in hBOA come mainly due to variance reduction. The main positive effect of variance reduction is the decrease in the required population size because of easier decision making and model building. The reduction in population sizes translates into the reduction in the number of evaluations and running time. Nonetheless, in some scenarios, variance reduction may also hurt performance of global-local hybrids by reducing diversity too much.

# Acknowledgments

This project was sponsored by the National Science Foundation under CAREER grant ECS-0547013, by the Air Force Office of Scientific Research, Air Force Materiel Command, 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. Experiments were done using the hBOA software developed by Martin Pelikan and David E. Goldberg at the University of Illinois at Urbana-Champaign and most experiments were performed on the Beowulf cluster maintained by ITS at the University of Missouri in St. Louis.

## References

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

Aickelin, U., Burke, E. K., & Li, J. (2007). An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*.

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

Davis, L. (Ed.) (1991). *Handbook of genetic algorithms*. New York, NY: Van Nostrand Reinhold.

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

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

Fischer, K., & Hertz, J. (1991). *Spin glasses*. Cambridge: Cambridge University Press.

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.

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

Goldberg, D. E., & Voessner, S. (1999). *Optimizing global-local search hybrids* (IlliGAL Report No. 99001). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

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

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

Harik, G. R., & Goldberg, D. E. (1996). Learning linkage. *Foundations of Genetic Algorithms*, *4*, 247–262.

Hart, W. E. (1994). *Adaptive global optimization with local search*. Doctoral dissertation, University of California, San Diego, San Diego, CA.

Hauschild, M., Pelikan, M., Lima, C., & Sastry, K. (2007). *Analyzing probabilistic models in hierarchical boa on traps and spin glasses* (MEDAL Report No. 2007001). Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri–St. Louis, St. Louis, MO.

Hoos, H. H., & Stützle, T. (2000). SATLIB: An online resource for research on SAT. (pp. 283–292). IOS Press.

Ibaraki, T. (1997). *Combinations with other optimization methods* (Chapter D3. Oxford University Press and Institute of Physics.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E., & Thatcher, J. W. (Eds.), *R. E. Miller and J. W. Thatcher* (pp. 85–103).

Land, M. (1998). *Adaptive global optimization with local search*. Doctoral dissertation, University of California, San Diego, San Diego, CA.

Lima, C., Pelikan, M., Sastry, K., Butz, M., & Goldberg, D. E. (2006). *Substructural neighborhoods for local search in the bayesian optimization algorithm* (MEDAL Report No. 2006007). Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri–St. Louis, St. Louis, MO.

Mezard, M., Parisi, G., & Virasoro, M. (1987). *Spin glass theory and beyond*. Singapore: World Scientific.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms.

Mühlenbein, H., & Mahnig, T. (2002). Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, *31*.

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

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. (2005). *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer-Verlag.

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. (2003). *Hierarchical BOA solves ising spin glasses and MAXSAT* (IlliGAL Report No. 2003001). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

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

Pelikan, M., & Hartmann, A. K. (2006). Searching for ground states of ising spin glasses with hierarchical BOA and cluster exact approximation. In Pelikan, M., Sastry, K., & Cantú-Paz, E. (Eds.), *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications* (pp. 333–349). Springer.

Pelikan, M., Katzgraber, H. G., & Kobe, S. (2008). *Finding ground states of sherrington-kirkpatrick spin glasses with hierarchical boa and genetic algorithms* (MEDAL Report No. 2008004). Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri–St. Louis, St. Louis, MO.

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

Sastry, K. (2001a). *Efficient atomic cluster optimization using a hybrid extended compact genetic algorithm with seeded population* (IlliGAL Report No. 2001018). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Sastry, K. (2001b). *Evaluation-relaxation schemes for genetic and evolutionary algorithms*. Master's thesis, University of Illinois at Urbana-Champaign.

Sinha, A., & Goldberg, D. E. (2001). *Verification and extension of the theory of global-local hybrids* (IlliGAL Report No. 2001010). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Sinha, A., & Goldberg, D. E. (2003). *A survey of hybrid genetic and evolutionary algorithms* (IlliGAL Report No. 2003004). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Thierens, D. (1999). Scalability problems of simple genetic algorithms. *Evolutionary Computation*, *7*(4), 331–352.

Thierens, D., & Goldberg, D. E. (1993). Mixing in genetic algorithms. pp. 38–45.

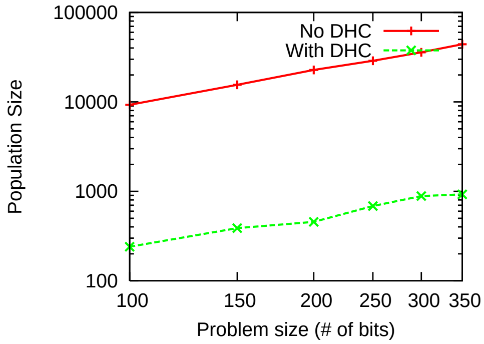
Tsutsui, S., Pelikan, M., & Ghosh, A. (2006). Edge histogram based sampling with local search for solving permutation problems. *International Journal of Hybrid Intelligent Systems*, *3*(1), 11–22.

Young, A. (Ed.) (1998). *Spin glasses and random fields*. Singapore: World Scientific.

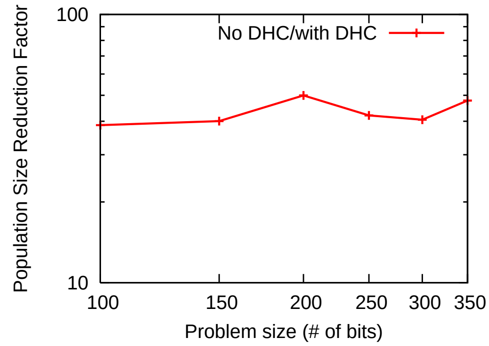
Yu, T.-L., Sastry, K., Goldberg, D. E., & Pelikan, M. (2007). Population sizing for entropy-based model building in estimation of distribution algorithms. pp. 601–608.

Zhang, Q., Sun, J., Tsang, E., & Ford, J. (2004). Hybrid estimation of distribution algorithm for global optimization. *Engineering Computations*, *21*(1), 91–107.

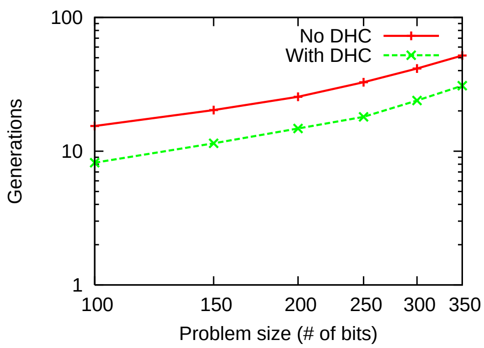
Zhang, Q., Sun, J., Tsang, E., & Ford, J. (2006). Estimation of distribution algorithm with 2-opt local search for the quadratic assignment problem. In *Towards a New Evolutionary Computation* (pp. 281–292). Springer.



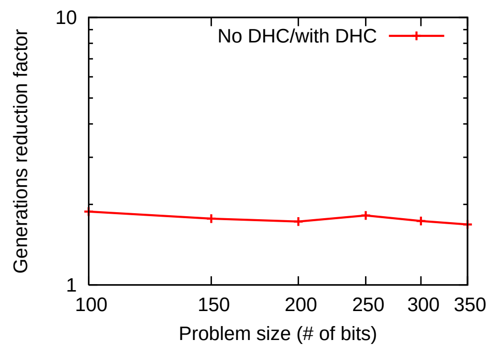

(a) Population sizes for trap-5.



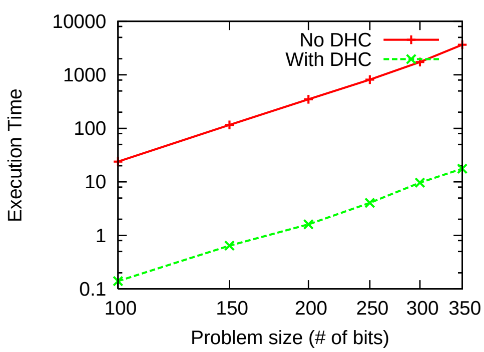

(b) Population size reduction for trap-5.



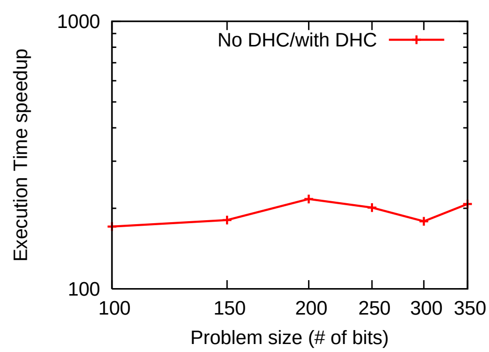

(c) Number of generations for trap-5.



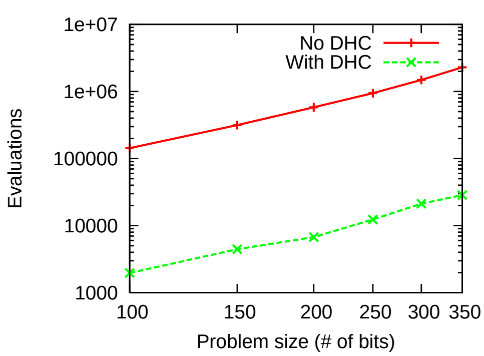

(d) Reduction in the number of generations for trap-5.



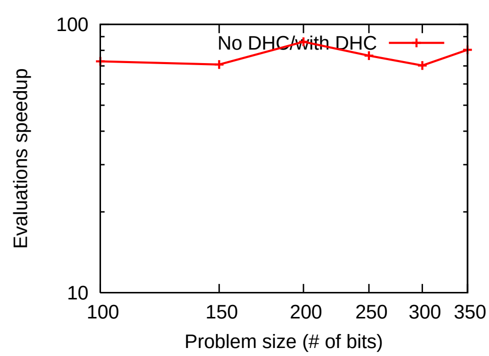

(e) Execution times for trap-5.




(f) Execution time speedups for trap-5.

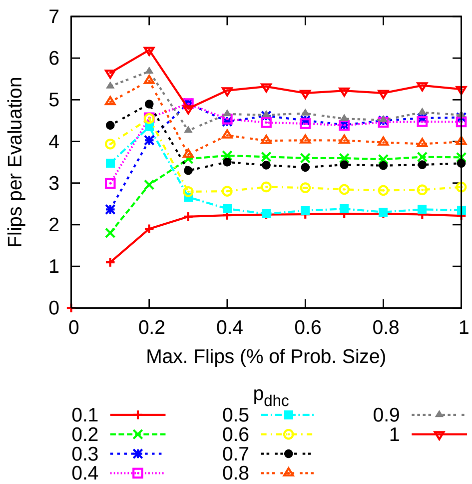



(g) Number of fitness evaluations for trap-5.

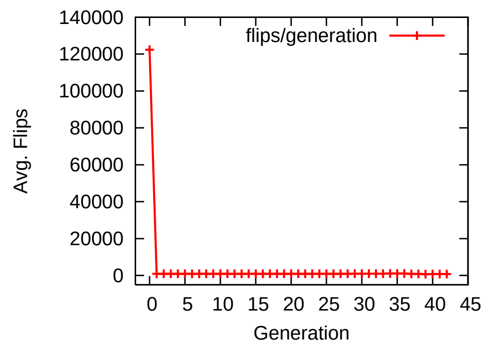



(h) Reduction in the number of fitness evaluations for trap-5.

Figure 3: These graphs compare the population sizes, numbers of generations, execution times, numbers of evaluations, and the factor by which these were reduced by incorporating DHC into hBOA on trap-5. More specifically, two cases are compared. In the first case, hBOA without DHC is used; in the second case, DHC is applied to 100% of the population and it is run until no more improvement is possible.

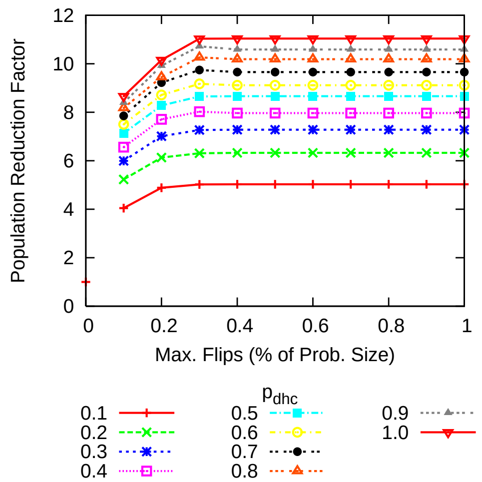



(a) Average number of flips performed per fitness evaluation for 350-bit trap-5.

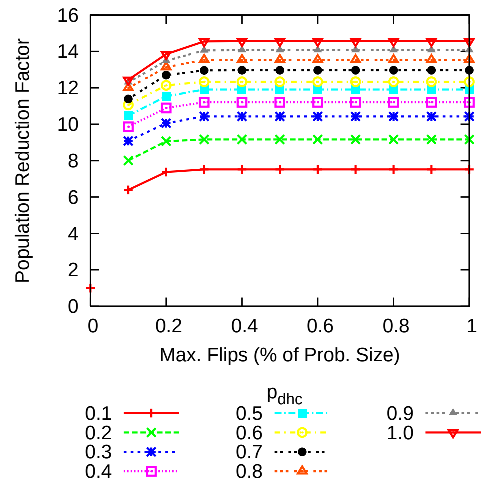



(b) Average number of flips performed per hBOA generation for 350-bit trap-5.

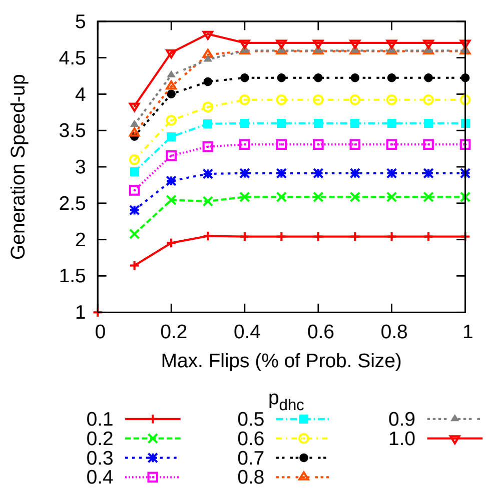
Figure 4: Part (a) of this figure shows the average number of flips per fitness evaluation for trap-5 of 350 bits. Part (b) shows the average number of flips performed in each generation of hBOA on the same problem. All results were obtained from averaging 10 independent bisections of 10 runs each.



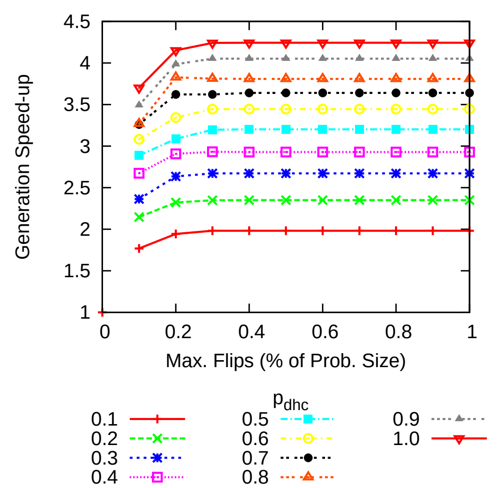

(a) Population size reduction for spin glass, $10 \times 10$.




(b) Population size reduction for spin glass, $16 \times 16$.

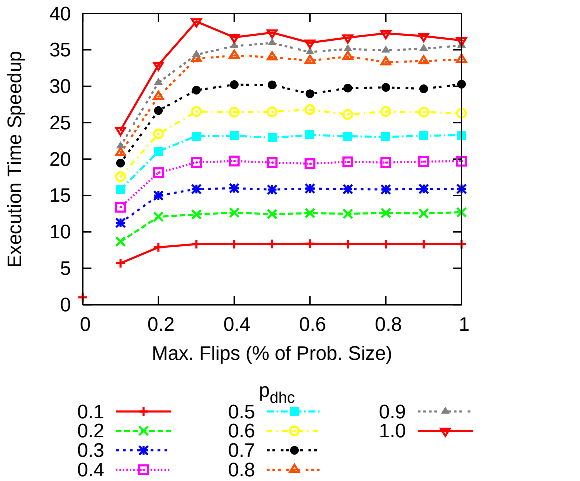



(c) Reduction in the number of generations for spin glass, $10 \times 10$.

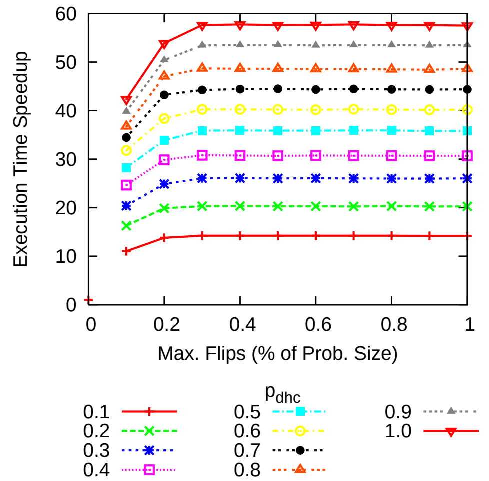



(d) Reduction in the number of generations for spin glass, $16 \times 16$.

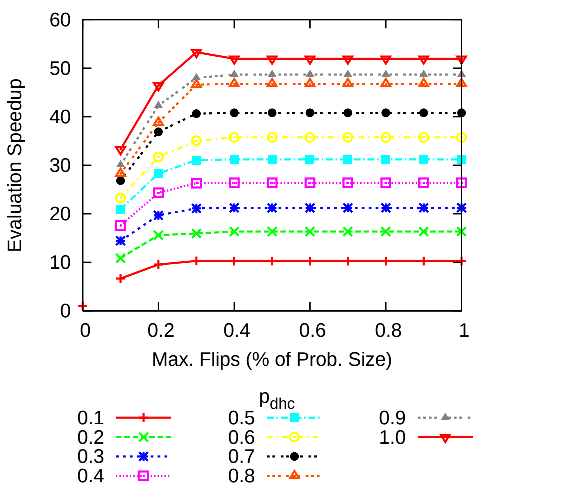
Figure 5: These graphs show the factor by which the population size and the number of generations were reduced by incorporating DHC into hBOA on the 2D Ising spin glass. Results for $10 \times 10$ (100 spins) and $16 \times 16$ (256 spins) are illustrated here; improvements were similar for all problem sizes tested ($8 \times 8$ - $16 \times 16$).



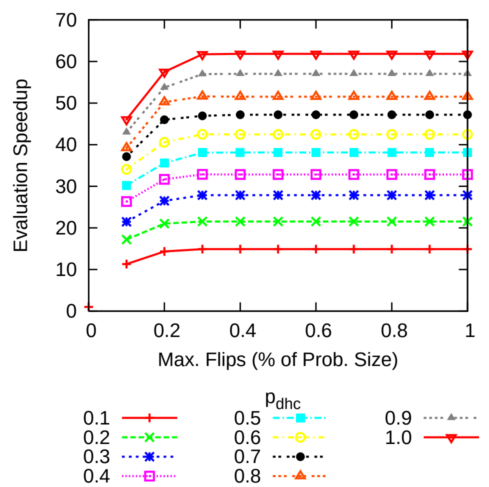

(a) Execution time speedups for spin glass, $10 \times 10$.




(b) Execution time speedups for spin glass, $16 \times 16$.

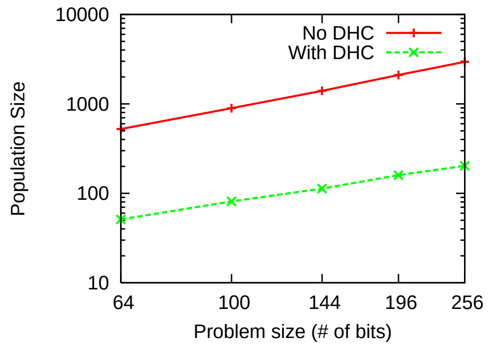



(c) Reduction in the number of fitness evaluations for spin glass, $10 \times 10$.

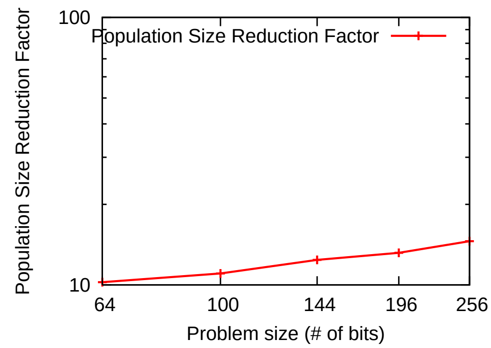



(d) Reduction in the number of fitness evaluations for spin glass, $16 \times 16$.

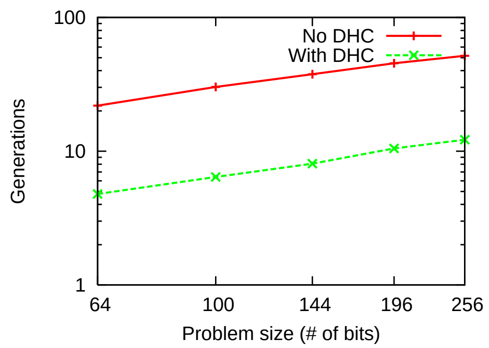
Figure 6: Improvements in execution times and the number of evaluations for $10 \times 10$ and $16 \times 16$ spin glasses are shown here. Similar improvements were seen for other problem sizes.



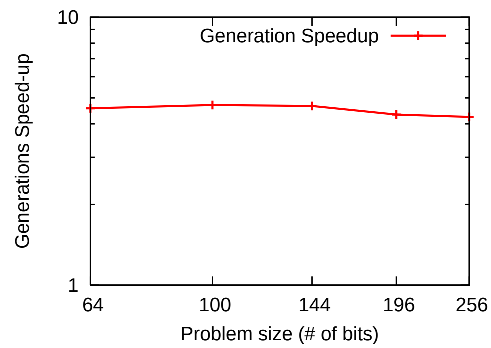

(a) Population sizes for spin glass.



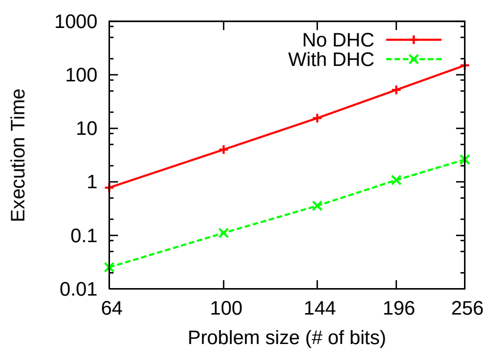

(b) Population size reduction for spin glass.



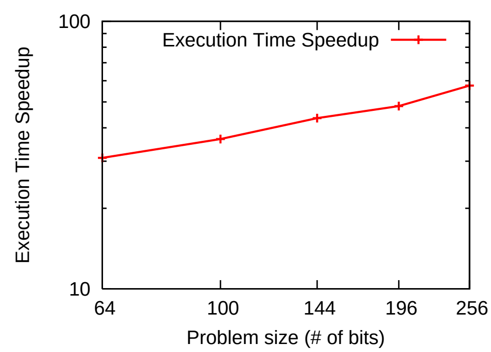

(c) Number of generations for spin glass.



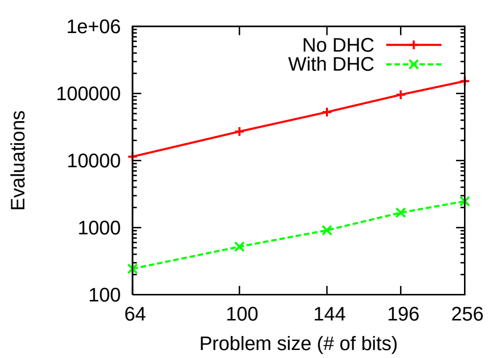

(d) Reduction in the number of generations for spin glass.



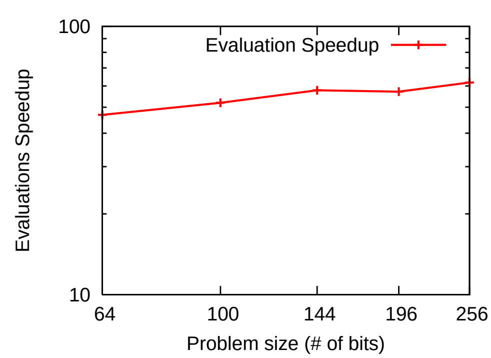

(e) Execution times for spin glass.




(f) Execution time speedups for spin glass.

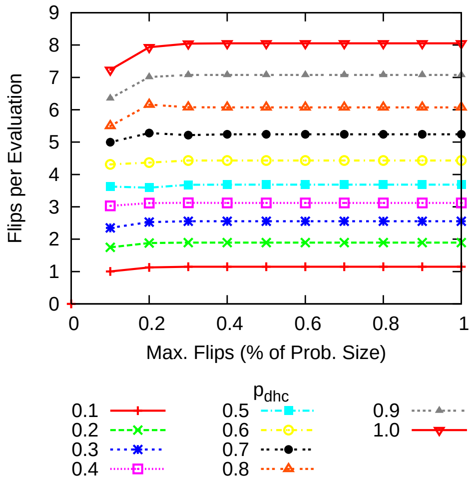



(g) Number of fitness evaluations for spin glass.

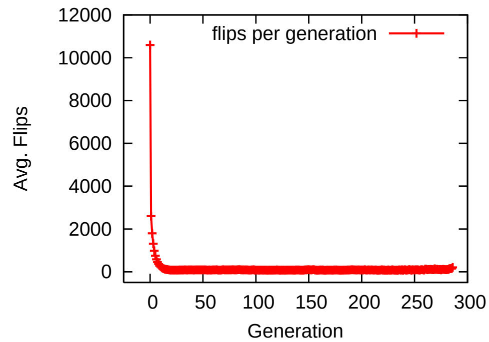



(h) Reduction in the number of fitness evaluations for spin glass.

Figure 7: These graphs compare the population sizes, numbers of generations, execution times, numbers of evaluations, and the factor by which these were reduced by incorporating DHC into hBOA on 2D Ising spin glass. More specifically, two cases are compared. In the first case, hBOA without DHC is used; in the second case, DHC is applied to 100% of the population and it is run until no more improvement is possible.

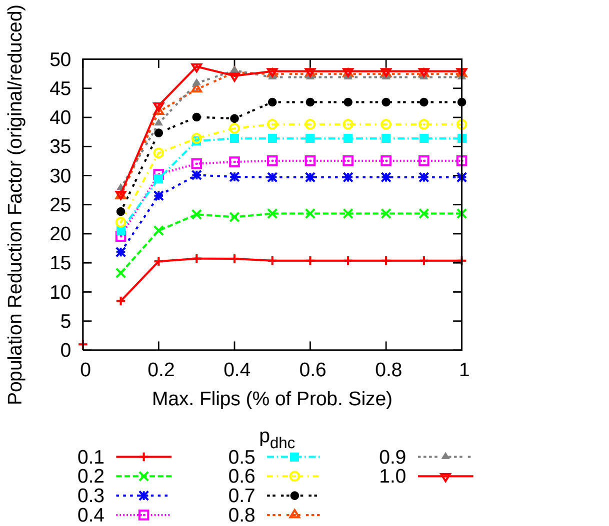



(a) Average number of flips performed per fitness evaluation for $16 \times 16$ spin glass.

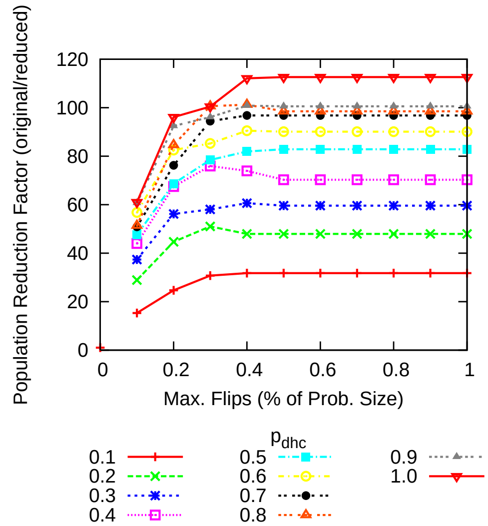



(b) Average number of flips performed per hBOA generation for $16 \times 16$ spin glass.

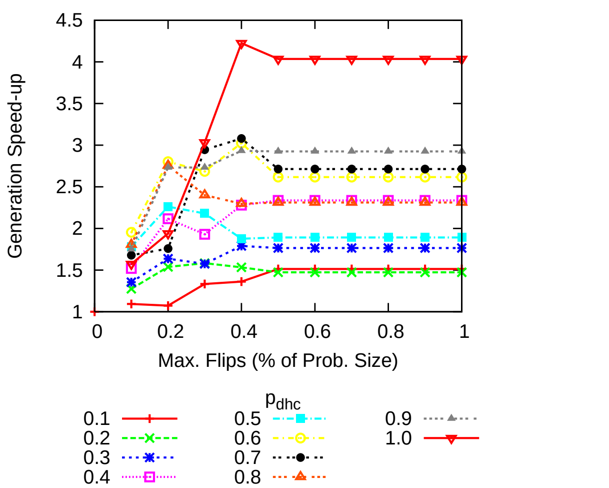
Figure 8: Part (a) of this figure shows the average number of flips per fitness evaluation for 2D spin glasses of size $16 \times 16$. Part (b) shows the average number of flips performed in each generation of hBOA on the same problem instances.



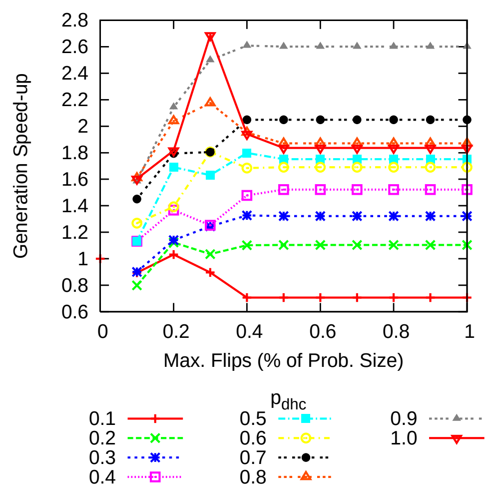

(a) Population size reduction for MAXSAT, 50 variables.




(b) Population size reduction for MAXSAT, 75 variables.

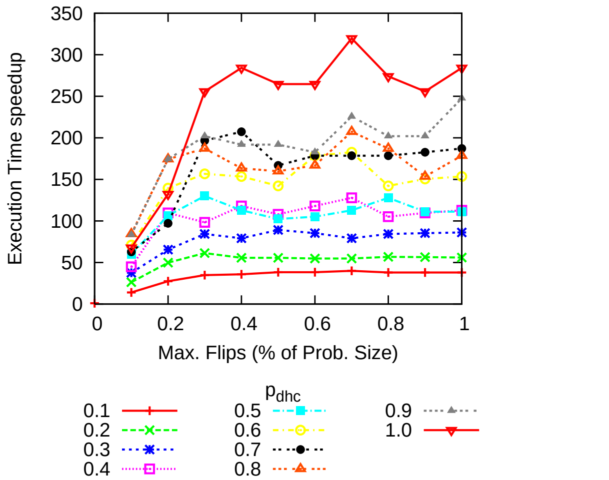



(c) Reduction in the number of generations for MAXSAT, 50 variables.

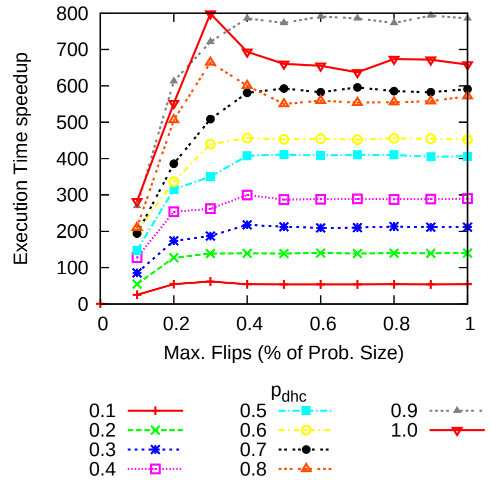



(d) Reduction in the number of generations for MAXSAT, 75 variables.

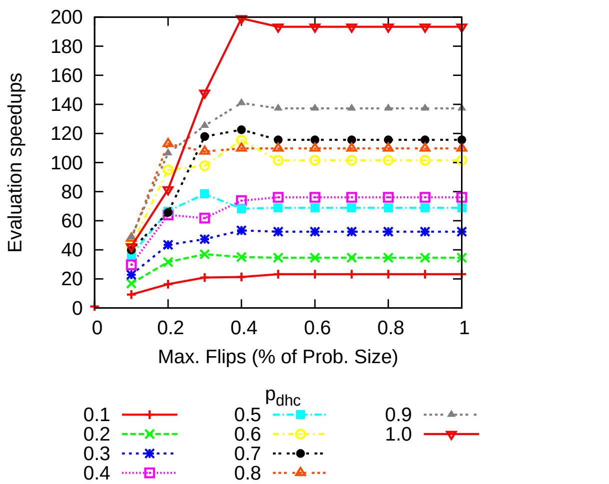
Figure 9: These figures show the factor by which the population size and the number of generations were reduced by using DHC in hBOA on uniform random 3-SAT. Results for 50 variables and 218 clauses, and for 75 variables and 325 clauses, are illustrated here.



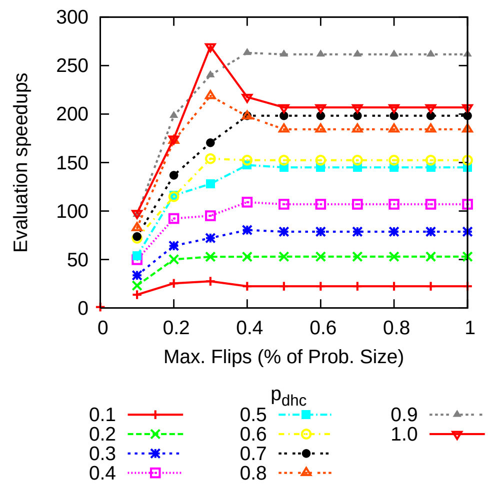

(a) Execution time speedups for MAXSAT, 50 variables.




(b) Execution time speedups for MAXSAT, 75 variables.

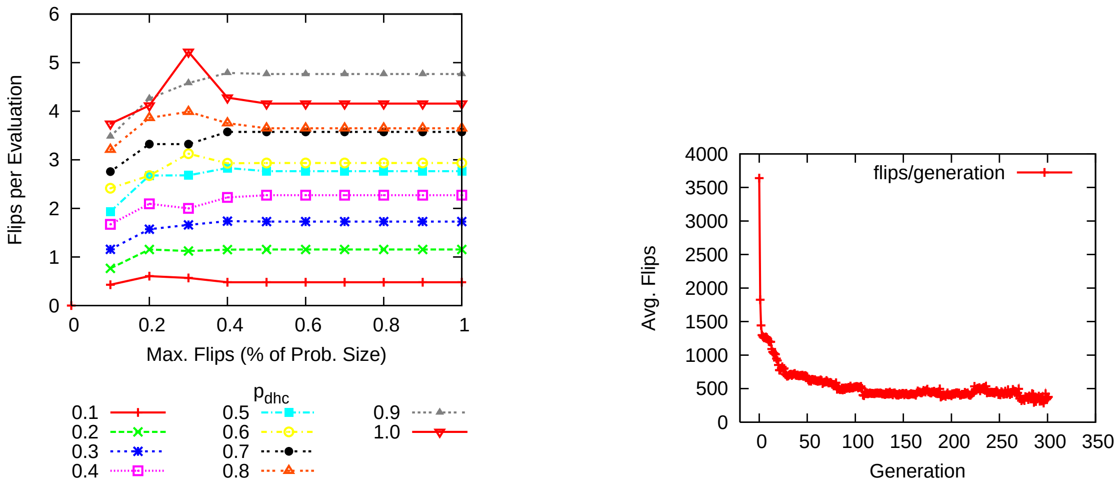



(c) Reduction in the number of fitness evaluations for MAXSAT, 50 variables.




(d) Reduction in the number of fitness evaluations for MAXSAT, 75 variables.

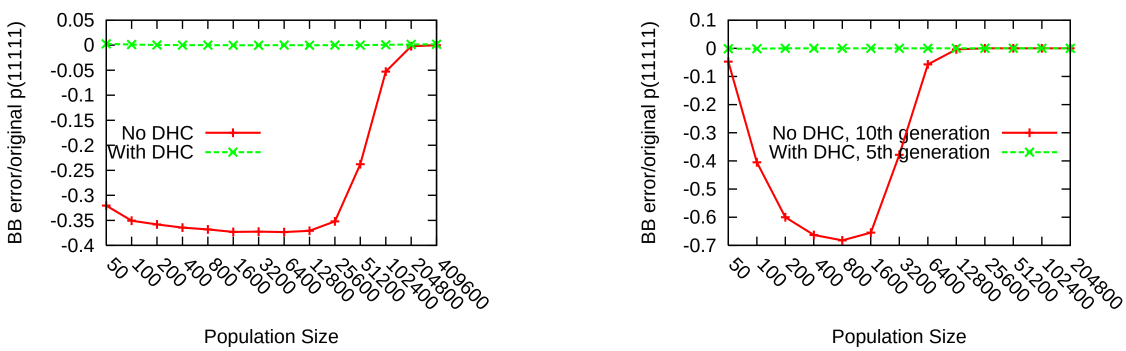
Figure 10: Improvements in execution time and the number of evaluations for MAXSAT of 50 and 75 variables are shown here.




(a) Average number of flips performed per fitness evaluation for 75 bit MAXSAT.

(b) Average number of flips performed per hBOA generation for 75 bit MAXSAT.

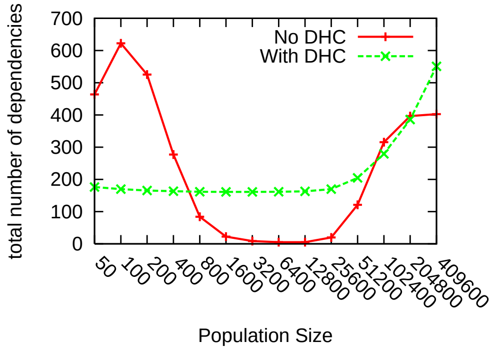
Figure 11: Part (a) of this figure shows the average number of flips per fitness evaluation for MAXSAT with 75 variables. Part (b) shows the average number of flips performed in each generation of hBOA on the same problem. All results were obtained from averaging 10 successful runs on each of the 100 MAXSAT instances.



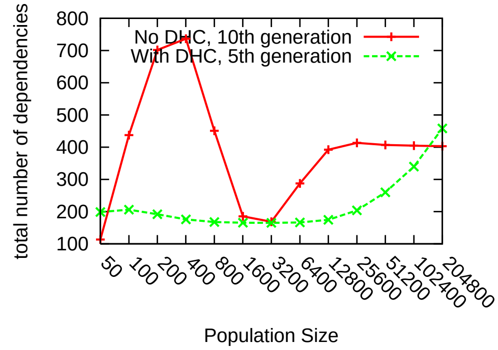

(a) BB loss in first generation

(b) BB loss from approximately the middle of the run (5th generation with DHC, 10th generation without DHC)

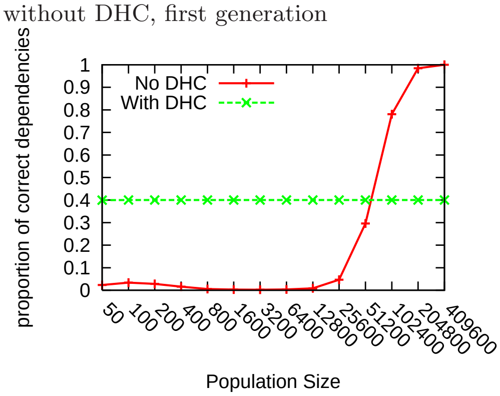
Figure 12: The loss in the proportion of optimal building blocks 11111 for the 200-bit trap-5 is visualized here as a function of the population size. The loss is expressed relative to the original number of copies of the building block; therefore, the loss of $-0.05$ represents the loss of 5% of the original number of copies before recombination. The loss directly relates to model accuracy—for a correct model, the optimal building blocks should be preserved. The figure considers two cases. In the first case, hBOA without DHC is used; in the second case, DHC is applied to 100% of the population and it is run until no more improvement is possible.



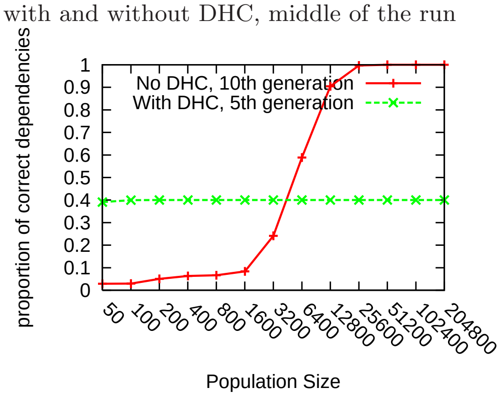

(a) Total number of dependencies discovered with and without DHC, first generation



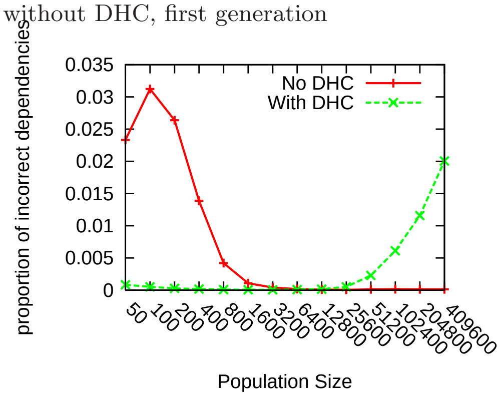

(b) Total number of dependencies discovered with and without DHC, middle of the run



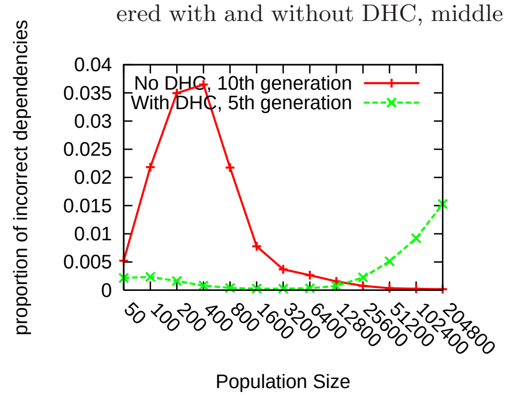

(c) Proportion of correct dependencies discovered with and without DHC, first generation




(d) Proportion of correct dependencies discovered with and without DHC, middle of the run

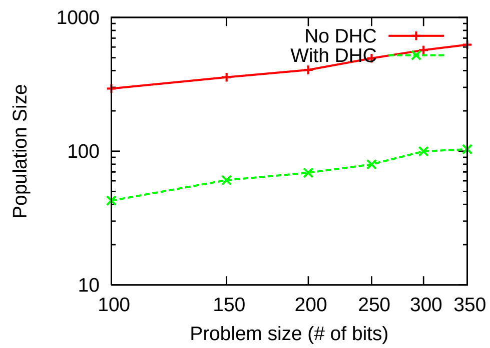



(e) Proportion of incorrect (spurious) dependencies discovered with and without DHC in the first generation.

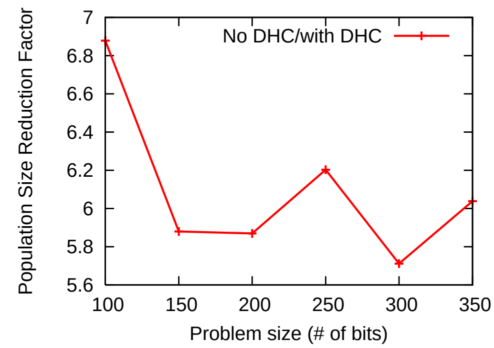



(f) Proportion of incorrect (spurious) dependencies discovered with and without DHC in the middle of the run.

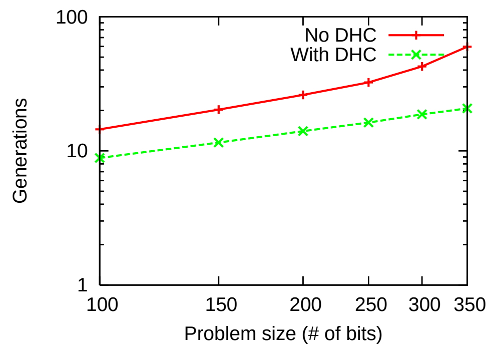
Figure 13: This figure visualizes model accuracy via the number of correct and spurious dependencies discovered by hBOA with and without DHC on 200-bit trap-5. In the case with DHC, DHC is applied to 100% of the population and is allowed to run until reaching a local optimum. To provide a better overview, the figure considers the first generation as well as the middle of the run. Parts (a) and (b) show the total number of dependencies in the model (correct and incorrect). Parts (c) and (d) show the percentages of correct dependencies discovered; 4 dependencies (40%) are necessary to at least connect all bits in the trap partition (with DHC this is actually enough), whereas 10 dependencies (100%) are the maximum possible (perfect model). Parts (e) and (f) show the percentage of spurious dependencies discovered out of all possible dependencies which could be generated. All results are averaged over 100 separate runs.



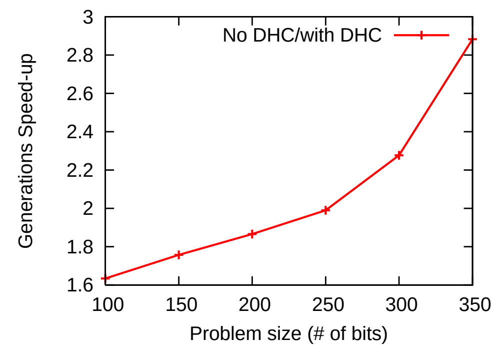

(a) Population sizes for trap-5, Perfect Model.



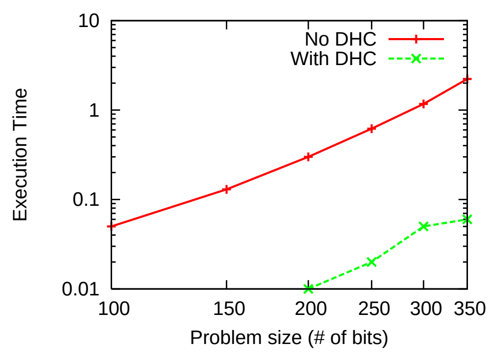

(b) Population size reduction for trap-5, Perfect Model.



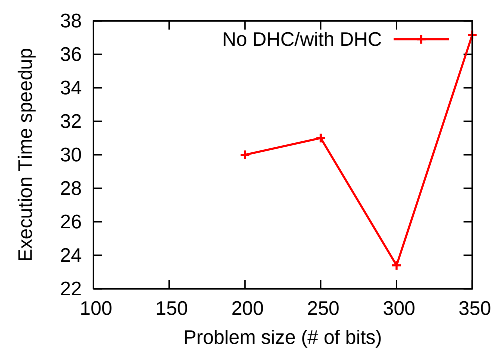

(c) Number of generations for trap-5, Perfect Model.



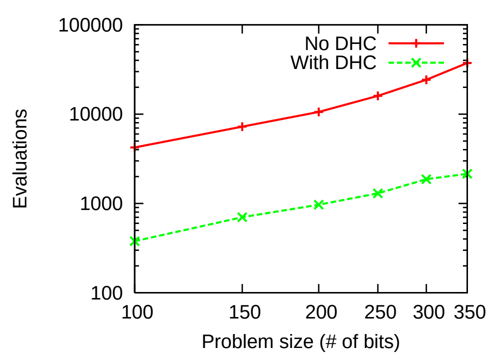

(d) Reduction in the number of generations for trap-5, Perfect Model.



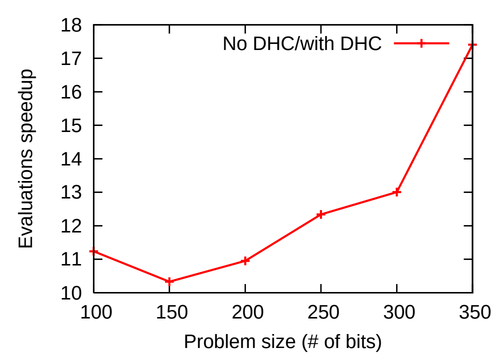

(e) Execution times for trap-5, Perfect Model.




(f) Execution time speedups for trap-5, Perfect Model.




(g) Number of fitness evaluations for trap-5, Perfect Model.




(h) Reduction in the number of fitness evaluations for trap-5, Perfect Model.

Figure 14: These results illustrate the effects of DHC when a perfect model for trap-5 is assumed. The graphs compare the population sizes, numbers of generations, execution times, numbers of evaluations, and the factor by which these were reduced by using DHC in hBOA compared to standard hBOA without DHC. In the case with DHC, DHC was applied to 100% of the population and it was run until no more improvement was possible.