Analysis of Epistasis Correlation on NK Landscapes with Nearest-Neighbor Interactions

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

ABSTRACT

Epistasis correlation is a measure that estimates the strength of interactions between problem variables. This paper presents an empirical study of epistasis correlation on a large number of random problem instances of NK landscapes with nearest neighbor interactions. The results are analyzed with respect to the performance of hybrid variants of two evolutionary algorithms: (1) the genetic algorithm with uniform crossover and (2) the hierarchical Bayesian optimization algorithm.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms, Performance

Keywords

Epistasis, epistasis correlation, problem difficulty, NK landscapes, genetic algorithms, estimation of distribution algorithms, linkage learning

1. INTRODUCTION

It has long been recognized that optimization problems with strong interactions between problem variables are often more difficult for genetic and evolutionary algorithms (GEAs) than problems where variables are nearly independent [8, 3, 5, 39, 24]. The strength of interactions between problem variables is often referred to as *epistasis*, a term used in biology to denote the amount of interaction between different genes. A number of approaches have been developed to adapt operators of GEAs to tackle problems with strong epistasis; these include for example linkage learning genetic algorithms [31, 13] and estimation of distribution algorithms (EDAs) [17, 27, 18, 28].

Two direct approaches were developed to measure the amount of epistasis in an optimization problem or the absence of it: (1) epistasis variance [3] and (2) epistasis correlation [34]. Epistasis correlation is often considered more useful, because its range is $[0, 1]$ and it is invariant with respect to linear transformations of fitness; the results may thus often be easier to interpret and compare. Because epistasis is strongly related to problem difficulty, measuring epistasis should provide insight into the difficulty of a problem. Nonetheless, it has been also recognized that a problem with strong epistasis is not necessarily more difficult than a problem with weaker epistasis [34, 21]. Although there are numerous papers discussing epistasis and measures of epistasis in the context of genetic and evolutionary algorithms [3, 4, 32, 34, 21], in most of these studies only a handful of problems are considered.

This paper presents a detailed empirical study of the relationship between problem parameters, the epistasis correlation, and the performance of two qualitatively different hybrid evolutionary algorithms, the genetic algorithm with uniform crossover (GA) [14, 8, 38], and the hierarchical Bayesian optimization algorithm (hBOA) [23]. In GA with uniform crossover, variation operators do not take into account correlations between variables and treat all variables as independent. On the other hand, hBOA can learn linkage; it is able to identify and exploit interactions between problem variables. Both GA and hBOA use hill climbing based on the single-bit neighborhood to speed up convergence and reduce computational requirements. As the target class of problems, the paper considers NK landscapes with nearest-neighbor interactions [24]. This problem class was chosen mainly because it provides a straightforward mechanism for tuning problem difficulty and level of epistasis, and it allows generation of a large number of random problem instances with known optima.

The paper is organized as follows. Section 2 describes epistasis variance and epistasis correlation, which are the two primary direct measures of epistasis in optimization. Section 3 describes the algorithms GA and hBOA, and the class of NK landscapes with nearest-neighbor interactions. Section 4 presents and discusses the experiments. Finally, section 5 summarizes and concludes the paper.

2. EPISTASIS

For success in both applied and theoretical research in evolutionary computation it is important to understand what makes one problem more difficult than another. Several ap-

GECCO'11, July 12–16, 2011, Dublin, Ireland.

proaches have been proposed to measure problem difficulty for evolutionary algorithms and other metaheuristics. The most popular measures include the fitness distance correlation [15], the autocorrelation function [41], the epistasis correlation [34], the signal-to-noise ratio [10], and scaling [40]. While many of these measures are related to epistasis, this paper focuses on approaches that measure epistasis directly.

In the remainder of this paper, candidate solutions are represented by binary strings of fixed length $n > 0$, although many of the discussed concepts and methods can be extended to other alphabets in a straightforward manner.

2.1 Epistasis Variance

This section describes *epistasis variance*, which is a measure of epistasis proposed by Davidor [3] and is defined as the Euclidean distance between the linear approximation of the fitness function and the actual fitness function over the population of all admissible solutions. To make the computation of epistasis variance tractable for moderate to large string length, we reduce the computation of the epistasis variance to an arbitrary population of candidate solutions.

Assume a population P of N candidate solutions represented by n-bit binary strings. The average fitness of solutions in P is defined as

$$\overline{f}(P) = \frac{1}{N}\sum_{x\in P} f(x).$$

Let us define the set of solutions in P with the value v_i in ith position as $P_i(v_i)$ and their number by $N_i(v_i)$. Then, for each position i in a solution string, we may define the fitness contribution of a bit v_i as

$$f_i(v_i) = \frac{1}{N_i(v_i)}\sum_{x\in P_i(v_i)} f(x) - \overline{f}(P) \tag{1}$$

The linear approximation of f [3, 32, 22, 33] is defined as

$$f_{lin}(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} f_i(X_i) + \overline{f}(P). \tag{2}$$

It is of note that the above linear fitness approximation has also been used in the first approach to modeling the fitness function in estimation of distribution algorithms [36]. The epistasis variance of f for population P is then defined as [3, 4]

$$\xi_P(f) = \sqrt{\frac{1}{N}\sum_{x\in P}\left(f(x) - f_{lin}(x)\right)^2} \tag{3}$$

One of the problems with epistasis variance is that its value changes even when the fitness function is just multiplied by a constant. That is why several researchers have proposed to normalize the epistasis variance, for example by dividing it by the variance of the fitness function [19], or by using a normalized fitness function [22].

2.2 Epistasis Correlation

Epistasis correlation was proposed by Rochet et al. [34] as a measure of epistasis that is invariant with respect to linear transformation of the fitness function (not only multiplication by a constant). Let us define the sum of square differences between f and $\overline{f}(P)$ over all members of P as

$$s_P(f) = \sum_{x\in P}\left(f(x) - \overline{f}(P)\right)^2.$$

Analogously, we may define the sum of square differences between f_{lin} and its average over P as

$$s_P(f_{lin}) = \sum_{x\in P}\left(f_{lin}(x) - \overline{f_{lin}}(P)\right)^2$$

where

$$\overline{f_{lin}}(P) = \frac{1}{N}\sum_{x\in P} f_{lin}(x).$$

The epistasis correlation for the population P is then defined as

$$epic_P(f) = \frac{\sum_{x\in P}\left(f(x) - \overline{f}(P)\right)\left(f_{lin}(x) - \overline{f_{lin}}(P)\right)}{s_P(f)s_P(f_{lin})} \tag{4}$$

The main advantage of epistasis correlation is that it is invariant with respect to linear transformations of fitness and its range is $[0, 1]$. Consequently, epistasis correlation is much easier to interpret than epistasis variance. These are the reasons why we use epistasis correlation in the remainder of this paper as the measure of epistasis.

2.3 Epistasis Measures and Difficulty

One of the problems with the epistasis correlation and the epistasis variance is that while the goal of these measures is to evaluate problem difficulty, the gap between problem difficulty and the epistasis measures is quite substantial. For example, if the epistasis correlation is 1, then we know that the problem is linear and it should thus be relatively easy to solve with practically any optimization method. Nonetheless, as the epistasis correlation decreases, this measure alone cannot be used as a single input to estimate problem difficulty because problem difficulty does not depend only on the *presence* of epistasis but also on its *character*. This observation was pointed out in many studies on epistasis variance or epistasis correlation, for example in references [34, 21].

The weakness of the connection between problem difficulty and the measures of epistasis has led to other models of problem difficulty originating in interactions between problem variables, such as *deception* [8, 9] and *fluctuating crosstalk* [37, 9]. One of the difficulties with these models is that it is not straightforward to quantify them in practice.

Nonetheless, it is of note that the weakness of the connection between the epistasis measures and problem difficulty is most often discussed on artificial problems that have little to do with the real world and that were created for the sole purpose of pointing out drawbacks of epistasis measures. In this paper, we aim to analyze the epistasis measures and their relationship to problem difficulty on a broad class of structured random problems, including both the easy and the difficult instances, that have much in common with important classes of real-world problems and that are often used as models of complex fitness landscapes.

2.4 Approximating Epistasis Correlation

Calculating the exact value of the epistasis correlation using a population of all possible strings is intractable for moderate to large values of n. Furthermore, approximating the value of epistasis correlation turns out to be slightly more challenging than approximating the values of some other measures of problem difficulty, such as the fitness distance correlation and the correlation length [24]. Since in this paper we considered 250,000 problem instances for which we computed the value of epistasis correlation, it was crucial to ensure that the computation of epistasis correlation is computationally efficient. Of course, for the results to be useful, accuracy was just as important as efficiency.

To estimate the value of the epistasis analysis, we started with $n_{exp} = 10$ independent experiments. In each of these experiments, we generated a population of 10^6 solutions, and we computed the exact value of the epistasis correlation for the generated population. The resulting epistasis correlation values were averaged to compute the final estimate of the epistasis correlation. If the results of the individual experiments indicated that the error in the average epistasis correlation value is away from its target value by more than 0.1% (assuming the normal distribution of the results), another 10 experiments were executed. Regardless of the error, the maximum number of experiments was 1,000. That means that the resulting estimate was computed from 10^7 to 10^9 independently generated samples.

3. PROBLEMS AND METHODS

This section outlines the algorithms and fitness functions used in the experiments.

3.1 Algorithms

3.1.1 Genetic algorithm

The genetic algorithm (GA) [14, 8] evolves a population of candidate solutions typically represented by binary strings of fixed length. The initial population is generated at random according to the uniform distribution over all binary strings. Each iteration starts by selecting promising solutions from the current population; we use binary tournament selection without replacement. New solutions are created by applying variation operators to the population of selected solutions. Specifically, crossover is used to exchange bits and pieces between pairs of candidate solutions and mutation is used to perturb the resulting solutions. Here we use uniform crossover [38], and bit-flip mutation [8]. To maintain useful diversity in the population, the new candidate solutions are incorporated into the original population using restricted tournament selection (RTS) [11]. The run is terminated when termination criteria are met. In this paper, each run is terminated either when the global optimum has been found or when a maximum number of iterations has been reached.

3.1.2 Hierarchical BOA

The hierarchical Bayesian optimization algorithm (hBOA) [25, 26, 23] is an estimation of distribution algorithm (EDA) [1, 20, 17, 27, 18, 28]. EDAs differ from GAs by replacing standard variation operators of GAs such as crossover and mutation by building a probabilistic model of promising solutions and sampling the built model to generate new candidate solutions. The only difference between GA and hBOA variants used in this study is that instead of using crossover and mutation to create new candidate solutions, hBOA learns a Bayesian network with local structures [2, 6] as a model of the selected solutions and generates new candidate solutions from the distribution encoded by this model. For more details on hBOA, see refs. [25, 23].

It is important to note that by building and sampling Bayesian networks, hBOA is able to scalably solve even problems with high levels of epistasis, assuming that the order of subproblems in an adequate problem decomposition is upper bounded by a constant [29]. Since the variation operators of the GA variant studied here assume that the string positions are independent whereas hBOA has a mechanism to deal with epistasis, it should be interesting to look at the effects of epistasis on these two algorithms. This is in fact the main reason for the choice of these two algorithms. In this context, an EDA based on univariate models may have been an even better choice than the GA with uniform crossover, but in that case most NK landscapes of larger size became intractable.

3.1.3 Bit-flip hill climber

The deterministic hill climber (DHC) is incorporated into both GA and hBOA to improve their performance similarly as in previous studies on using GA and hBOA for solving NK landscapes and related problems [30, 24]. DHC takes a candidate solution represented by an n-bit binary string on input. Then, it performs one-bit changes on the solution that lead to the maximum improvement of solution quality. DHC is terminated when no single-bit flip improves solution quality and the solution is thus locally optimal. Here, DHC is used to improve every solution in the population before the evaluation is performed.

3.2 Nearest-neighbor NK landscapes

NK fitness landscapes [16] were introduced by Kauffman as tunable models of rugged fitness landscape. An NK fitness landscape is fully defined by the following components: (1) The number of bits, n, (2) the number of neighbors per bit, k, (3) a set of k neighbors $\Pi(X_i)$ for the i-th bit for every $i \in \{1, \ldots, n\}$, and (4) a subfunction f_i defining a real value for each combination of values of X_i and $\Pi(X_i)$ for every $i \in \{1, \ldots, n\}$. Typically, each subfunction is defined as a lookup table. The objective function f_{nk} to maximize is defined as

$$f_{nk}(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} f_i(X_i, \Pi(X_i)).$$

In this paper, we consider nearest-neighbor NK landscapes, in which neighbors of each bit are restricted to the k bits that immediately follow this bit. The neighborhoods wrap around; thus, for bits that do not have k bits to the right, the neighborhood is completed with the first few bits of solution strings. The reason for restricting neighborhoods to nearest neighbors was to ensure that the problem instances can be solved in polynomial time even for $k > 1$ using dynamic programming [24]. The subfunctions are represented by look-up tables (a unique value is used for each instance of a bit and its neighbors), and each entry in the look-up table is generated with the uniform distribution from $[0, 1)$. To make the problem more difficult for conventional variation operators based on tight linkage between bits located close to each other, the string positions are randomly shuffled prior to optimization. The used class of NK landscapes with nearest neighbors is thus the same as that in ref. [24].

In this paper, we consider $k \in \{2, 3, 4, 5, 6\}$ and $n = 20$ to 100 with step 10 (for scalability experiments) or 20 (for epistasis correlation). For each combination of n and k, we generated and tested 10,000 unique problem instances. The reason for using such a large number of instances was to get a sufficient number of samples for the various tests presented here. For GA, the results for instances with $n = 100$ and

$k = 6$ were too computationally expensive and they were thus omitted. In summary, for epistasis correlation, 250,000 independently generated problem instances were used and 450,000 independently generated instances were used for experiments on scalability of hBOA and GA.

The difficulty of optimizing NK landscapes depends on all components defining an NK problem instance [42]. Although NK landscapes with nearest neighbor interactions are polynomially solvable in terms of n [24], the difficulty of problem instances from this class generally increases as n and k grow. On nearest-neighbor NK landscapes, the time complexity of most evolutionary algorithms is expected to grow at least polynomially fast with n, and no better than exponentially fast with k [30, 24].

4. EXPERIMENTS

4.1 hBOA and GA Parameter Settings

In hBOA Bayesian networks with decision trees [7, 2] are used as probabilistic models. To evaluate models, the Bayesian-Dirichlet metric with likelihood equivalence [2] and the penalty for model complexity [23] is used. In GA, uniform crossover and bit-flip mutation are used as variation operators. The probability of crossover is $p_c = 0.6$ and the probability of flipping a bit with mutation is $1/n$ where n is the number of bits. To select promising solutions, binary tournament selection without replacement is used in both GA and hBOA. New solutions are incorporated into the original population using RTS [11] with window size $w = \min\{n, N/20\}$ as suggested by ref. [23]. The population sizes are identified using bisection [35, 23] to ensure convergence in 10 out of 10 independent runs. Each run is terminated either when the global optimum is found (success) or when the maximum number of iterations equal to the number of bits n has been reached (failure). The maximum number of iterations was n based on preliminary experiments which indicated that if a run did not find the optimum after n iterations, it was unlikely to find the optimum or finding it would take extremely long time.

4.2 Performance of hBOA and GA

Before presenting the results for the epistasis correlation, let us examine performance of hBOA and GA hybrids with respect to the values of n and k. Figure 1 shows the growth of the number of evaluations and the number of local search steps (DHC flips) with n and k for both hybrids. Note that each step of local search may require several full or partial evaluations, depending on the problem. For nearest-neighbor NK landscapes, the complexity of each step of local search can be upper bounded by $O(\log n)$, whereas each evaluation takes $O(n)$ time. That is why we distinguish between the number of evaluations and the number of flips.

The results confirm that the number of evaluations and the number of flips grow polynomially fast with the number n of bits. The results also confirm that both these statistics grow at least exponentially fast with k.

4.3 Effects of Problem Size and Neighborhood Size on Epistasis Correlation

Figure 2 shows scatter plots of epistasis correlation with respect to the number of evaluations and the number of steps of local search (DHC flips) for $n = 100$ and $k \in \{2, 3, 4, 5, 6\}$. The figure indicates that, as expected, as k grows, epistasis increases in magnitude and, in agreement with this, the epistasis correlation decreases with k. This is confirmed with the results shown in figure 3b, which also considers $n = 100$ and $k \in \{2, 3, 4, 5, 6\}$. It is of note that although figures 2 and 3b consider only one problem size $n = 100$, the results for other problem sizes are nearly identical.

Scatter plots in figure 2 also indicate that the number of evaluations and the number of steps of the local searcher vary more for GA than for hBOA; this is expected because uniform crossover cannot deal with epistasis and assumes that the variables are independent.

Figure 3a shows that the epistasis correlation does not seem to be affected by the overall number of bits in the problem, because it stays nearly constant regardless of the problem size. This observation is not a surprise; problem difficulty originating in problem size is related mainly to collateral noise [9, 12], whereas epistasis is concerned only with the amount of interaction between problem variables.

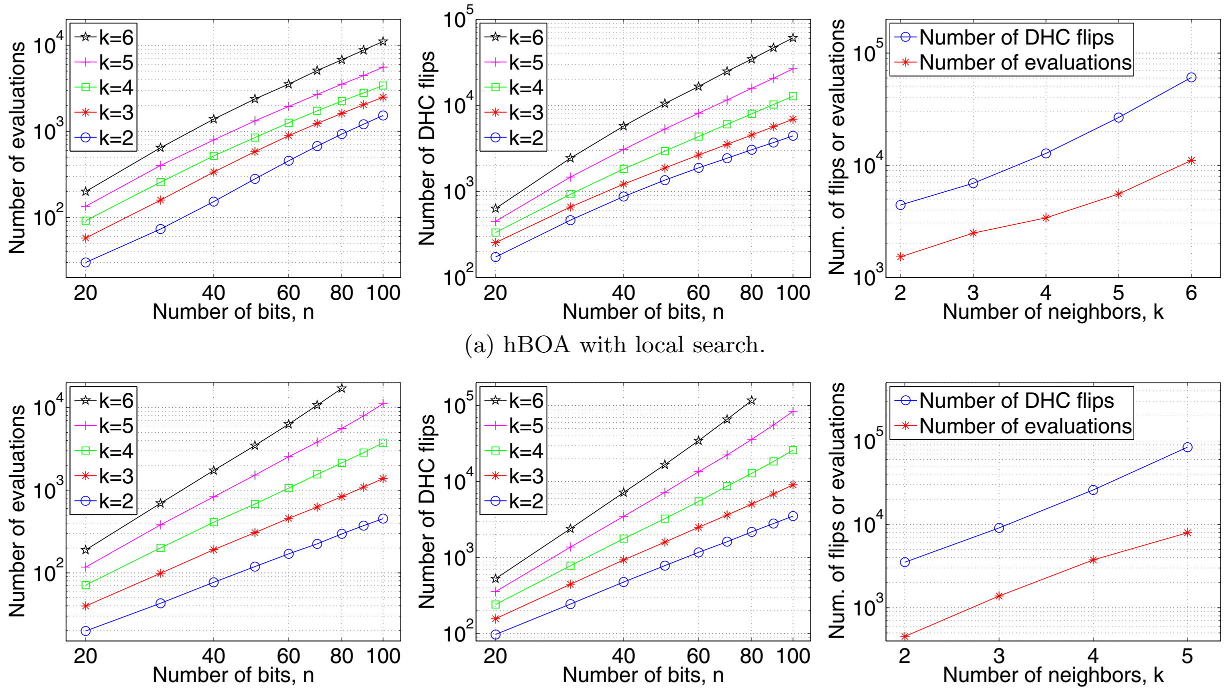
4.4 Epistasis Correlation and Problem Difficulty with Fixed Problem Size and Neighborhood Size

While the relationship between epistasis correlation, n and k was in agreement with our understanding of epistasis and the difficulty of NK landscapes, it is not as clear what the relationship between epistasis and problem difficulty is for fixed n and k. In this section, we examine the relationship between epistasis correlation and the actual difficulty of problem instances assuming fixed n and k.

Specifically, for various combinations of n and k, several subsets of easy and difficult problem instances are selected, and for each of these subsets, the average number of steps of local search and the average epistasis correlation are presented. As subsets, we select the 10%, 25% and 50% easiest instances, the 10%, 25% and 50% hardest instances, and all instances for the specific combination of n and k regardless of their difficulty. Since for each combination of n and k, 10,000 instances were used, even the smallest subset of 10% instances contains 1,000 instances. The difficulty of an instance is measured by the actual number of steps of local search using the optimization method under consideration (either GA with uniform crossover and DHC, or hBOA with DHC). In most cases, $n = 100$ and $k \in \{2, 3, 4, 5, 6\}$. The results for other problem sizes are similar. However, for GA with uniform crossover, resource constraints did not allow us to complete experiments $k = 6$ and $n = 90$ or $n = 100$, so we used $n = 80$ for $k = 6$. The results are shown in tables 1 and 2.

As shown in table 1, for hBOA with DHC and most values of k, the values of epistasis correlation are in agreement with the actual difficulty of the subsets of instances. This is in agreement with our intuition that as the problem difficulty increases, the epistasis correlation decreases, indicating an increased level of epistasis. Nonetheless, as k grows, the *differences* between the values of epistasis correlation for the different subsets of instances decrease. In fact, for $k = 5$ and $k = 6$, the values of epistasis correlation sometimes *increase* with problem difficulty. While these results are somewhat surprising, they can be explained by the fact that hBOA is able to deal with problems with epistasis of bounded order efficiently. That is why hBOA should not be as sensitive to epistasis as many other evolutionary algorithms.

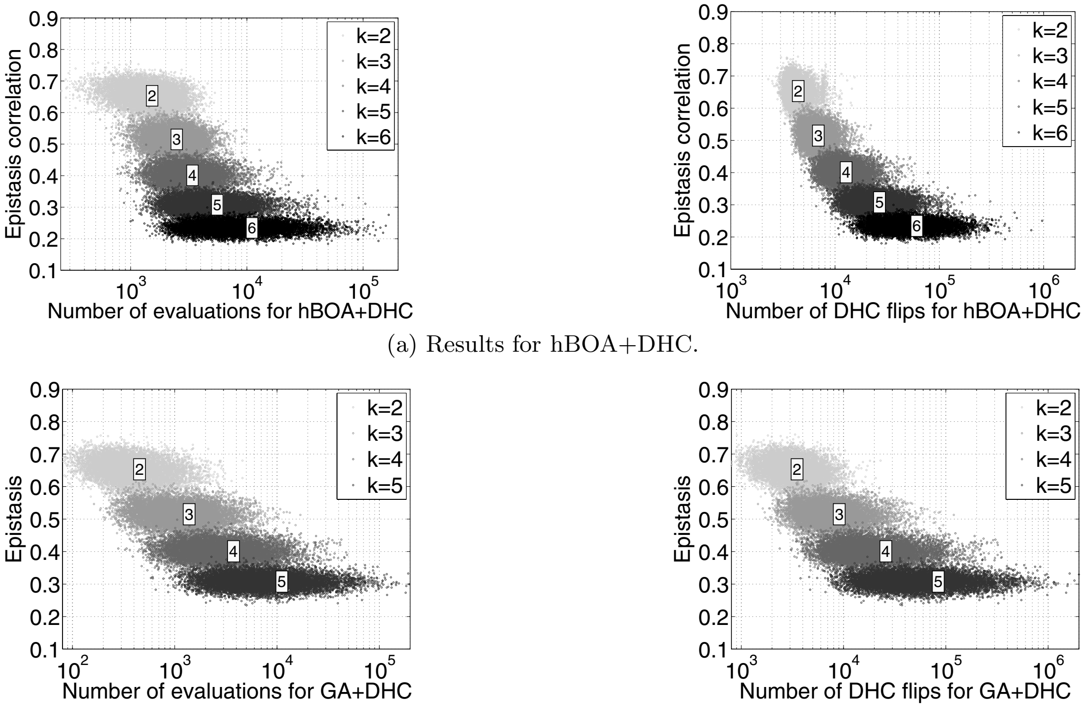
As shown in table 2, for GA with uniform crossover and DHC, the values of epistasis correlation are also in agreement



(a) hBOA with local search.

(b) GA with uniform crossover and local search.

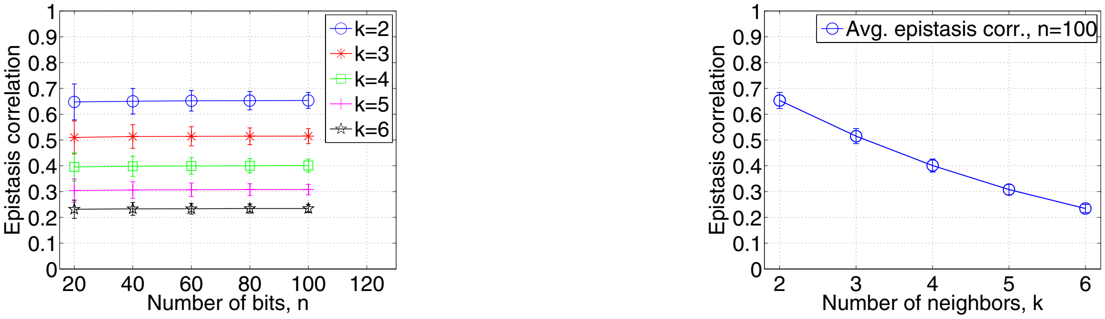
Figure 1: Performance of hBOA and GA with uniform crossover on nearest-neighbor NK landscapes of $k = 2$ to $k = 6$ and $n = 20$ to $n = 100$.



(a) Results for hBOA+DHC.

(b) Results for GA+DHC.

Figure 2: Scatter plots of the epistasis correlation with respect to the number of evaluations and the number of steps of the local search (DHC flips) for GA and hBOA.



(a) Epistasis correlation with respect to n.

(b) Epistasis correlation with respect to k.

Figure 3: Epistasis correlation with respect to the number n of bits and the number k of neighbors of nearest-neighbor NK landscapes.

with our understanding of how epistasis affects problem difficulty. As the problem instances become more difficult, the epistasis correlation decreases, indicating an increased level of epistasis. In fact, for GA, the results are in agreement with our understanding of epistasis and problem difficulty even for larger values of k, although the differences between the epistasis values in different subsets decrease with k.

The differences between the results for hBOA and GA confirm that the effect of epistasis should be weaker for hBOA than for GA because hBOA can deal with epistasis better than conventional GAs by detecting and using interactions between problem variables. The differences are certainly small, but so are the differences between the epistasis correlation values between the subsets of problems that are even orders of magnitude different in terms of the computational time. The differences between a conventional GA with no linkage learning and one of the most advanced EDAs are among the most interesting results in this paper.

5. SUMMARY AND CONCLUSIONS

This paper discussed epistasis and its relationship with problem difficulty. To measure epistasis, epistasis correlation was used. The empirical analysis considered hybrids of two qualitatively different evolutionary algorithms and a large number of instances of nearest-neighbor NK landscapes.

The use of epistasis correlation in assessing problem difficulty has received a lot of criticism [22, 34]. The main reason for this is that although the absence of epistasis does imply that a problem is easy, the presence of epistasis does not necessarily imply that the problem is difficult. Nonetheless, given our current understanding of problem difficulty, there is no doubt that introducing epistasis increases the *potential* of a problem to be difficult.

This paper indicated that for randomly generated NK landscapes with nearest-neighbor interactions, epistasis correlation correctly captures the fact that the problem instances become more difficult as the order of interactions (number of neighbors) increases. Additionally, the results confirmed that for a fixed problem size and order of interactions, sets of more difficult problem instances have lower values of epistasis correlation (and, thus, stronger epistasis). The results indicated also that evolutionary algorithms capable of linkage learning are less sensitive to epistasis than conventional evolutionary algorithms.

The bad news is that the results confirmed that epistasis correlation does not provide a single input for the practitioner to assess problem difficulty, even if we assume that the problem size and the order of interactions are fixed, and all instances are generated from the same distribution. In many cases, simple problems included strong epistasis and hard problems included weak epistasis. A similar observation has been made in ref. [24] for the correlation length and the fitness distance correlation. However, compared to these other popular measures of problem difficulty, epistasis correlation belongs to one of the more accurate ones, at least for the class of randomly generated NK landscapes with nearest-neighbor interactions.

One important topic for future work is to compile some of the past results in analysis of various measures of problem difficulty with the results presented here, and explore the ways in which different measures of problem difficulty can be combined to provide a problem-difficulty measure that outperforms any of the individual measures. The experimental study presented in this paper should also be extended to other classes of problems, especially those that allow one to generate a large set of random problem instances. Classes of spin glass optimization problems and graph problems are good candidates for these efforts. Finally, one may consider the variants of the presented algorithms without local search, although in that case, performance of all algorithms deteriorates significantly and the analysis may have to be performed only for problem instances of much smaller size.

Acknowledgments

This project was sponsored by the National Science Foundation under CAREER grant ECS-0547013, and by the Univ. of Missouri in St. Louis through the High Performance Computing Collaboratory sponsored by Information Technology Services. Most experiments were performed on the Beowulf cluster maintained by ITS at the Univ. of Missouri in St. Louis. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

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

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

(a) hBOA, $n = 100$, $k = 2$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	3330.9 (163.9)	0.6645 (0.030)
25% easiest	3550.2 (217.0)	0.6608 (0.030)
50% easiest	3758.6 (265.2)	0.6580 (0.030)
all instances	4436.2 (1019.5)	0.6534 (0.031)
50% hardest	5113.8 (1044.2)	0.6487 (0.031)
25% hardest	5805.5 (1089.4)	0.6466 (0.031)
10% hardest	6767.6 (1152.3)	0.6447 (0.032)

(b) hBOA, $n = 100$, $k = 3$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	4643.5 (264.0)	0.5221 (0.029)
25% easiest	5071.2 (420.4)	0.5193 (0.029)
50% easiest	5618.8 (651.1)	0.5175 (0.029)
all instances	6919.3 (1795.8)	0.5150 (0.029)
50% hardest	8219.7 (1626.0)	0.5124 (0.029)
25% hardest	9298.2 (1690.0)	0.5109 (0.029)
10% hardest	10688.3 (1925.0)	0.5106 (0.028)

(c) hBOA, $n = 100$, $k = 4$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	7380.6 (689.2)	0.4049 (0.024)
25% easiest	8361.4 (958.0)	0.4031 (0.025)
50% easiest	9528.3 (1414.2)	0.4025 (0.025)
all instances	12782.4 (4897.7)	0.4009 (0.025)
50% hardest	16036.5 (4979.6)	0.3994 (0.025)
25% hardest	19203.1 (5384.0)	0.3990 (0.025)
10% hardest	23674.1 (6088.7)	0.3986 (0.025)

(d) hBOA, $n = 100$, $k = 5$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	12526.4 (1408.4)	0.3090 (0.020)
25% easiest	14662.1 (2078.3)	0.3085 (0.020)
50% easiest	17399.8 (3334.6)	0.3085 (0.020)
all instances	26684.2 (14255.6)	0.3079 (0.020)
50% hardest	35968.5 (14931.0)	0.3072 (0.020)
25% hardest	44928.0 (16730.7)	0.3068 (0.020)
10% hardest	58353.5 (19617.4)	0.3071 (0.020)

(e) hBOA, $n = 100$, $k = 6$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	21364.1 (2929.9)	0.2349 (0.016)
25% easiest	26787.7 (5261.6)	0.2351 (0.016)
50% easiest	34276.6 (8833.1)	0.2348 (0.016)
all instances	60774.8 (42442.8)	0.2344 (0.016)
50% hardest	87272.9 (46049.2)	0.2339 (0.016)
25% hardest	114418.9 (52085.3)	0.2340 (0.016)
10% hardest	154912.8 (62794.1)	0.2341 (0.016)

Table 1: Epistasis correlation for easy and hard instances for hBOA. The difficulty of instances is measured by the overall number of steps of the local searcher.

(a) GA (uniform), $n = 100$, $k = 2$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	1493.1 (259.4)	0.6660 (0.028)
25% easiest	1881.9 (373.9)	0.6625 (0.030)
50% easiest	2332.3 (543.2)	0.6588 (0.030)
all instances	3516.7 (1741.5)	0.6534 (0.031)
50% hardest	4701.0 (1722.0)	0.6479 (0.031)
25% hardest	5840.9 (1800.5)	0.6443 (0.031)
10% hardest	7358.3 (2002.9)	0.6395 (0.030)

(b) GA (uniform), $n = 100$, $k = 3$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	3240.8 (461.9)	0.5249 (0.029)
25% easiest	4035.6 (790.6)	0.5215 (0.029)
50% easiest	5178.5 (1340.6)	0.5189 (0.029)
all instances	9082.9 (6558.2)	0.5150 (0.029)
50% hardest	12987.4 (7330.5)	0.5110 (0.029)
25% hardest	17116.9 (8517.9)	0.5095 (0.029)
10% hardest	23829.5 (10164.2)	0.5082 (0.030)

(c) GA (uniform), $n = 100$, $k = 4$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	6594.6 (1183.6)	0.4045 (0.024)
25% easiest	8590.6 (1928.4)	0.4037 (0.025)
50% easiest	11445.0 (3427.8)	0.4026 (0.025)
all instances	25903.7 (26303.0)	0.4009 (0.025)
50% hardest	40362.3 (30885.2)	0.3993 (0.025)
25% hardest	57288.4 (36351.8)	0.3989 (0.025)
10% hardest	85279.2 (44200.4)	0.3970 (0.025)

(d) GA (uniform), $n = 100$, $k = 5$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	13898.4 (2852.6)	0.3099 (0.020)
25% easiest	19872.9 (5765.2)	0.3098 (0.020)
50% easiest	29259.2 (11063.4)	0.3087 (0.020)
all instances	84375.6 (119204.9)	0.3079 (0.020)
50% hardest	139492.0 (149074.5)	0.3070 (0.020)
25% hardest	209536.2 (185682.5)	0.3068 (0.020)
10% hardest	335718.7 (242644.0)	0.3058 (0.019)

(e) GA (uniform), $n = 80$, $k = 6$

desc. of instances	DHC steps until optimum	epistasis correlation
10% easiest	15208.7 (3718.2)	0.2358 (0.018)
25% easiest	22427.3 (6968.4)	0.2358 (0.018)
50% easiest	34855.5 (14722.9)	0.2353 (0.018)
all instances	117021.4 (204462.0)	0.2344 (0.018)
50% hardest	199187.4 (264378.2)	0.2335 (0.018)
25% hardest	310451.2 (338773.5)	0.2330 (0.018)
10% hardest	519430.9 (461122.7)	0.2324 (0.018)

Table 2: Epistasis correlation for easy and hard instances for GA with uniform crossover. The difficulty of instances is measured by the overall number of steps of the local searcher.

[3] Y. Davidor. Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383, 1990.

[4] Y. Davidor. *Genetic algorithms and robotics: a heuristic strategy for optimization.* World Scientific Publishing, Singapore, 1991.

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

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

[7] N. Friedman and Z. Yakhini. On the sample complexity of learning Bayesian networks. *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI-96)*, pages 274–282, 1996.

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

[9] D. E. Goldberg. *The design of innovation: Lessons from and for competent genetic algorithms.* Kluwer, 2002.

[10] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

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

[12] G. R. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proc. of the Int. Conf. on Evolutionary Computation (ICEC-97)*, pages 7–12, 1997.

[13] G. R. Harik and D. E. Goldberg. Learning linkage. *Foundations of Genetic Algorithms*, 4:247–262, 1996.

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

[15] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. *Proc. of the Int. Conf. on Genetic Algorithms (ICGA-95)*, pages 184–192, 1995.

[16] S. Kauffman. Adaptation on rugged fitness landscapes. In D. L. Stein, editor, *Lecture Notes in the Sciences of Complexity*, pages 527–618. Addison Wesley, 1989.

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

[18] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms.* Springer, 2006.

[19] M. Manela and J. A. Campbell. Harmonic analysis, epistasis and genetic algorithms. *Parallel Problem Solving from Nature*, pages 59–66, 1992.

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

[21] B. Naudts and L. Kallel. Some facts about so called GA-hardness measures. Technical Report 379, Ecole Polytechnique, CMAP, France, 1998.

[22] B. Naudts, D. Suys, and A. Verschoren. Epistasis as a basic concept in formal landscape analysis. *Proc. of the Int. Conf. on Genetic Alg. (ICGA-97)*, pages 65–72, 1997.

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

[24] M. Pelikan. NK landscapes, problem difficulty, and hybrid evolutionary algorithms. *Genetic and Evol. Comp. Conf. (GECCO-2010)*, pages 665–672, 2010.

[25] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. *Genetic and Evol. Comp. Conf. (GECCO-2001)*, pages 511–518, 2001.

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

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

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

[29] M. Pelikan, K. Sastry, and D. E. Goldberg. Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258, 2002.

[30] M. Pelikan, K. Sastry, D. E. Goldberg, M. V. Butz, and M. Hauschild. Performance of evolutionary algorithms on NK landscapes with nearest neighbor interactions and tunable overlap. *Genetic and Evol. Comp. Conf. (GECCO-2009)*, pages 851–858, 2009.

[31] Y. ping Chen, T.-L. Yu, K. Sastry, and D. E. Goldberg. A survey of genetic linkage learning techniques. IlliGAL Report No. 2007014, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2007.

[32] C. R. Reeves and C. C. Wright. Epistasis in genetic algorithms: An experimental design perspective. *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 217–224, 1995.

[33] S. Rochet, M. Slimane, and G. Venturini. Epistasis for real encoding in genetic algorithms. In *Proceedings of the Australian New Zealand Conference on Intelligent Information Systems, ANZIIS 96, Adelaide, South Australia, 18-20 November 1996*, pages 268–271, 1996.

[34] S. Rochet, G. Venturini, M. Slimane, and E. M. E. Kharoubi. A critical and empirical study of epistasis measures for predicting ga performances: A summary. In *Selected Papers from the Third European Conference on Artificial Evolution*, AE '97, pages 275–286, London, UK, 1998. Springer-Verlag.

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

[36] K. Sastry, D. E. Goldberg, and M. Pelikan. Don't evaluate, inherit. *Genetic and Evol. Comp. Conf. (GECCO-2001)*, pages 551–558, 2001.

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

[38] G. Syswerda. Uniform crossover in genetic algorithms. *Proc. of the Int. Conf. on Genetic Algorithms (ICGA-89)*, pages 2–9, 1989.

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

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

[41] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, 1990.

[42] A. H. Wright, R. K. Thompson, and J. Zhang. The computational complexity of N-K fitness functions. *IEEE Trans. on Evolutionary Computation*, 4(4):373–379, 2000.