Missouri Estimation of Distribution Algorithms Laboratory

NK Landscapes, Problem Difficulty, and Hybrid Evolutionary Algorithms

Martin Pelikan

MEDAL Report No. 2010001

January 2010

Abstract

This paper presents an empirical study of NK landscapes with the main focus on the relationship between (1) problem parameters, (2) various measures of problem difficulty of fitness landscapes, and (3) performance of hybrid evolutionary algorithms. As the target class of problems, the study considers two types of NK landscapes: (1) Standard, unrestricted NK landscapes and (2) shuffled NK landscapes with nearest-neighbor interactions. As problem difficulty measures, the paper considers the fitness distance correlation, the correlation coefficient, the distance of local and global optima, and the escape rate. Experimental results are presented, analyzed and discussed. Promising avenues for future work are also outlined.

Keywords

NK fitness landscape, problem difficulty, performance analysis, fitness distance correlation, correlation length, escape rate, hybrid evolutionary algorithms, estimation of distribution algorithms.

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/

NK Landscapes, Problem Difficulty, and Hybrid Evolutionary Algorithms

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

January 30, 2010

Abstract

This paper presents an empirical study of NK landscapes with the main focus on the relationship between (1) problem parameters, (2) various measures of problem difficulty of fitness landscapes, and (3) performance of hybrid evolutionary algorithms. As the target class of problems, the study considers two types of NK landscapes: (1) Standard, unrestricted NK landscapes and (2) shuffled NK landscapes with nearest-neighbor interactions. As problem difficulty measures, the paper considers the fitness distance correlation, the correlation coefficient, the distance of local and global optima, and the escape rate. Experimental results are presented, analyzed and discussed. Promising avenues for future work are also outlined.

Keywords: NK fitness landscape, problem difficulty, performance analysis, fitness distance correlation, correlation length, escape rate, hybrid evolutionary algorithms, estimation of distribution algorithms.

1 Introduction

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 approaches have been proposed to measure problem difficulty for evolutionary algorithms and other metaheuristics. The most popular measures include the fitness distance correlation [13], the autocorrelation function [29], the signal-to-noise ratio [9], and scaling [28]. A number of studies investigated these measures on various types of optimization problems [29, 13, 16, 27]. However, most of these studies considered only several isolated problem instances and only a handful of detailed studies exist that focus on large sets of randomly generated instances of important classes of problems [27, 24]. Yet, analysis on large sets of random instances is often immensely helpful in gaining a better understanding of the strengths and limitations of evolutionary algorithms and other optimization methods [21, 23, 27].

The purpose of this paper is to present a detailed empirical study of the relationship between problem parameters, problem difficulty measures, and performance of hybrid evolutionary algorithms. As the target class of problems, the study uses NK fitness landscapes [14, 15], which were introduced by Kauffman as tunable models of rugged fitness landscape. The paper considers two types of NK landscapes: (1) Standard, unrestricted NK landscapes and (2) shuffled NK landscapes with nearest-neighbor interactions. As problem difficulty measures, the paper considers the fitness distance correlation for arbitrary solutions or only the local optima, the correlation length, the distance of local and global optima, and the escape rate. As a representative evolutionary algorithm, a hybrid of the hierarchical Bayesian optimization algorithm (hBOA) [19, 20, 18] and local search is used. Hybrid hBOA was chosen mainly because it is capable of linkage learning, and it was shown to outperform conventional evolutionary algorithms on numerous classes of additively decomposable problems such as NK landscapes [23].

The paper starts by describing NK landscapes and the types of NK landscapes studied in this work. Section 3 outlines the problem difficulty measures. Section 4 describes the hierarchical BOA with local search, which is used as the main algorithm in the experimental part of the paper. Section 5 presents experimental results. Section 6 summarizes and concludes the paper, and discusses the future work.

2 NK Landscapes

This section describes NK landscapes and the specific types of NK landscapes studied in this work.

2.1 Problem Definition

An NK fitness landscape [14, 15] 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, X_i, for every $i \in \{0, \ldots, n-1\}$. (4) A subfunction g_i defining a real value for each combination of values of X_i and $\Pi(X_i)$ for every $i \in \{0, \ldots, n-1\}$. Typically, each subfunction is defined as a lookup table with 2^{k+1} values. The task is to maximize the objective function

$$f_{nk}(X_0, X_1, \ldots, X_{n-1}) = \sum_{i=0}^{n-1} g_i(X_i, \Pi(X_i)).$$

The difficulty of optimizing NK landscapes depends on all of the four components defining an NK problem instance. One useful approach to analyzing complexity of NK landscapes is to focus on the influence of k on problem complexity. For $k = 0$, NK landscapes are simple unimodal functions similar to onemax or binint, which can be solved in linear time and should be easy for practically any genetic and evolutionary algorithm. The global optimum of NK landscapes can be obtained in polynomial time [30] even for $k = 1$; on the other hand, for $k > 1$, the problem of finding the global optimum of unrestricted NK landscapes is NP-complete [30]. The problem becomes polynomially solvable with dynamic programming even for $k > 1$ if the neighbors are restricted to only adjacent string positions (using circular strings) [30] or if the subfunctions are generated according to some distributions [7]. For unrestricted NK landscapes with $k > 1$, a polynomial-time approximation algorithm exists with the approximation threshold $1 - 1/2^{k+1}$ [30].

Because of their difficulty, properties, and similarity with other classes of difficult optimization problems, NK landscapes have attracted researchers in a number of areas, especially in stochastic optimization and computational biology [15, 2, 30, 7, 1, 5].

2.2 Considered Classes of NK Landscapes

Problem instances in this work are inspired by two recent papers on performance analysis of evolutionary algorithms on NK landscapes [23, 24]. More specifically, two types of NK landscapes are considered: (1) Standard NK landscapes with randomly generated subfunctions and neighborhoods and (2) nearest-neighbor NK landscapes where the neighbors are restricted to the consequent positions in the solution strings (the neighborhoods wrap around at the end). In both cases, each subfunction is defined by a unique lookup table, the elements of which are generated at random using the uniform distribution over $[0, 1)$. For nearest-neighbor NK landscapes, the string positions are randomly shuffled to eliminate tight linkage, although for the algorithm considered here, the ordering of the variables does not matter.

The considered class of random problem instances of standard NK landscapes is identical as that in ref. [23]. The class of nearest-neighbor instances is similar to that in ref. [24], although here the neighborhoods wrap around whereas in ref. [24] the neighborhoods were cut at the end of the string.

2.3 Identifying Optima of NK Landscapes

In order to provide useful results and problem difficulty analyses, it is desirable to know the value and location of the global optima. Although NK landscapes are NP-complete for $k > 1$, a branch and bound algorithm can be used to solve relatively small instances of sizes up to 30 to 60 bits if neighborhood size is relatively small [23]. The nearest-neighbor instances can be solved using dynamic programming using an approach similar to that in ref. [24].

To solve standard NK instances, the branch and bound implementation obtained from the authors of ref. [23] was used. To solve nearest-neighbor NK instances, we modified the dynamic programming implementation for solving random additively decomposable problems [22]. It is beyond the scope of this paper to discuss details of these methods; for more information, please see refs. [22, 23, 24].

Due to the method used for generating problem instances, it is highly unlikely that there exist multiple global optima for any instance. This was verified for the nearest-neighbor instances, and there has not been a single generated instance that had more than one global optimum. Although it is computationally prohibitive to verify this fact for most instances of unrestricted NK landscapes, in the experiments presented here it is assumed that even in this case there is a single global optimum for each problem instance.

3 Measuring Problem Difficulty

A fitness landscape consists of three main components: (1) A set S of admissible solutions, (2) a fitness function f that assigns a real value to each solution in S, and (3) a distance measure d that defines a distance between any two solutions in S. S and f define the problem being solved. Specifically, the task is to find $\arg\max_{x \in S} f(x)$. On the other hand, the distance measure depends on the operators used. Specifically, $d(x, y)$ defines the number of steps to get from x to y. For binary strings, Hamming distance is often used, which is equal to the number of string positions in which the two binary strings differ. Hamming distance metric accurately represents distances between solutions using the simple neighborhood based on flipping a single bit at a time. For more complex variation operators, such as crossover, other distance measures may be more appropriate, although the standard bit-flip neighborhoods provide useful inputs on problem difficulty even in those cases.

The remainder of this section discusses some of the approaches for measuring the difficulty of fitness landscapes.

3.1 Fitness Distance Correlation

Consider a set of n candidate solutions with fitness values $F = \{f_1, f_2, \ldots, f_n\}$ and a corresponding set $D = \{d_1, d_2, \ldots, d_n\}$ of the distances of these solutions to the nearest global optimum. The *fitness distance correlation* (FDC) quantifies the strength and nature of the relationship between the fitness value and the distance to the nearest global optimum as

$$r = \frac{c_{FD}}{\sigma_F \sigma_D}, \tag{1}$$

where σ_F and σ_D are standard deviations of F and D, respectively, and c_{FD} is the covariance of F and D. The covariance c_{FD} is defined as

$$c_{FD} = \frac{1}{n} \sum_{i=1}^{n} (f_i - \bar{f})(d_i - \bar{d}),$$

where $\bar{f}$ and $\bar{d}$ are the means of F and D, respectively. Note that the computation of FDC necessitates knowledge of *all* global optima.

FDC takes values from $[-1, 1]$. Assuming that we are interested in maximization of fitness, it should be easier to find a global optimum for smaller values of FDC than for larger ones, because for small FDC values the fitness values point towards global optima more consistently than they do for high values. On the other hand, higher values of FDC indicate that the fitness may often mislead the search away from the global optimum. Thus, the smaller the values of r, the easier the maximization problem should be. For example, for onemax, $r = -1$, whereas for the fully deceptive trap function of size 20, $r \approx +1$ [13].

Since in this paper the main focus is on a hybrid evolutionary algorithm, we also consider a modification of FDC in which only local optima are used in the computation of FDC. In this variant, each value f_i corresponds to the fitness value of a local optimum and the distance d_i denotes the distance of this local optimum to the closest global one. We denote FDC restricted to local optima by r_l or FDCL.

3.2 Correlation Length

Consider a random walk through the landscape which starts in a random solution and moves to a random neighbor of the current solution in each step (neighbors of a candidate solution are all solutions at distance 1 from it). To measure problem difficulty based on random walks, we can use the *random walk correlation function* (also called the fitness autocorrelation function) [29], which quantifies the strength of the relationship between the fitness values of a candidate solution x and the solutions that are obtained by taking a given number s of steps starting in x. In other words, the correlation function quantifies ruggedness of the landscape.

For a random walk of $m - 1$ steps passing through solutions of fitness values $\{f_t\}_{t=1\ldots m}$, the random walk correlation function $\rho(s)$ for gap s is defined as [29]

$$\rho(s) = \frac{1}{\sigma_F^2 (m - s)} \sum_{t=1}^{m-s} \left(f_t - \bar{f}\right)\left(f_{t+s} - \bar{f}\right),$$

where s is the number of steps (gap), and $\bar{f}$ and σ_F denote the average fitness and the standard deviation of the fitness values, respectively. Typically, the larger the value of s, the weaker the correlations between fitness values; $\rho(s)$ can thus be expected to decrease with increasing s. Furthermore, the smaller the value of $\rho(s)$, the more rugged the landscape is. Therefore, the landscape should be relatively easier to explore for smaller $\rho(s)$ than for larger $\rho(s)$.

The correlation function can be used to compute the correlation length, which estimates the effective range of correlations between states in a random walk. The correlation length may be defined as [26]

$$l = -\frac{1}{\ln(|\rho(1)|)}.$$

The correlation function $\rho(s)$ can also be used to compute the autocorrelation coefficient $\delta = 1/(1-\rho(1))$ [3], which has approximately the same value as the correlation length [17]. The smaller the correlation length or autocorrelation coefficient, the harder the problem instance.

3.3 Escape Rate

One of the factors that may influence problem difficulty especially for hybrid algorithms that combine global and local search is the number of steps required to escape a local optimum. To quantify this factor, Merz [16] defined the *escape rate* measure. The escape rate for a given number s of steps is the probability of escaping the basin of attraction of a local optimum after performing s steps starting in this local optimum.

One way to estimate the escape rate is to consider a set of local optima $\{x_1, x_2, \ldots, x_n\}$ as starting points. Then, for each x_i, m solutions $S_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,m}\}$ are created that are at distance of s steps from x_i. For each created solution $y_{i,j}$, local search is executed to find the local optimum at the basin of attraction of $y_{i,j}$. The percentage of solutions from S_i that lead to a different local optimum than x_i is then defined as the escape rate.

The greater the escape rate for any particular value of s, the easier it is to escape a local optimum after performing s steps in the search space. From this perspective, the greater the escape rate, the easier the problem should be. Nonetheless, at the same time, greater escape rates also indicate a more rugged landscape, thereby increasing the difficulty. This is confirmed with the empirical results presented in this paper, which indicate that as the problems become more difficult to solve, the escape rates decrease.

3.4 Distance of Local and Global Optima

Additionally to the above measures of problem difficulty, we computed the average distance of local optima to the closest global optimum. The distance is measured by the number of steps of a local searcher (e.g., the number of mismatched bits). Although it is clearly advantageous when the local optima are closer to the global one, the results indicate that in some cases this measure is misleading and in fact shorter distances pose a greater challenge.

4 Hybrid Hierarchical BOA

As a representative evolutionary algorithm, a hybrid of the hierarchical Bayesian optimization algorithm (hBOA) [19, 20, 18] and the deterministic local search is used in this paper.

hBOA evolves a population of candidate solutions typically represented by binary strings of fixed length. The first population is generated at random according to the uniform distribution over all solutions. Each iteration starts by selecting promising solutions from the current population; here binary tournament selection without replacement is used. Next, hBOA (1) learns a Bayesian network with local structures for the selected solutions and (2) generates new candidate solutions by sampling the distribution encoded by the built network [4, 6]. To maintain useful diversity in the population, the new candidate solutions are incorporated into the original population using restricted tournament selection (RTS) [10]. The run is terminated when termination criteria are met. In this paper, each run is terminated either when the global optimum is found or when a maximum number of iterations is reached. For more details on hBOA, please consult refs. [19, 18].

The deterministic hill climber (DHC) is incorporated into hBOA to improve its performance. 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. Without DHC, the number and size of problem instances would have to substantially reduce due to the increased computational requirements.

5 Experiments

5.1 Problem Instances

For both types of NK landscapes, the number of neighbors ranged from $k = 2$ to $k = 6$ with step 1. For standard NK landscapes, problem sizes were limited due to the computational complexity of branch an bound; the problem sizes ranged from $n = 20$ to $n = 56$ for $k = 2$, $n = 48$ for $k = 3$, $n = 40$ for $k = 4$, $n = 40$ for $k = 5$, and $n = 36$ for $k = 6$. The problem size was increased with step of 4. For NK landscapes with nearest neighbors, the problem sizes ranged from $n = 20$ to $n = 100$ with step 10. For each considered combination of n and k, 10^4 unique problem instances were generated. For each of these, a guaranteed optimum was found using the branch and bound or dynamic programming.

5.2 hBOA Parameters

To select promising solutions, binary tournament selection without replacement was used. New solutions (offspring) were incorporated into the old population using RTS with window size $w = \min\{n, N/5\}$ as suggested in ref. [18]. Bayesian networks with decision trees [4, 6, 18] were used and the models were evaluated using the Bayesian-Dirichlet metric with likelihood equivalence [12, 4] and a penalty for model complexity [6, 18].

For each problem instance, an adequate population size was approximated using bisection [25, 18] to ensure that the optimum is found in 10 out of 10 independent runs. Each run was terminated when the global optimum was found (success) or when the maximum number of generations equal to the number of bits n was reached (failure). There are four relevant statistics that can be used to determine the overall complexity of hBOA: the population size, the number of evaluations, the number of DHC steps, and the number of generations (iterations). Due to the variety of hardware, CPU time is not as useful. According to the results, the number of DHC flips represents the overall complexity of the search most reliably and we thus use this statistic to quantify hBOA time requirements.

5.3 Estimating Measures of Difficulty

To estimate the fitness distance correlation r for each problem instance, 100 samples of 1,000 points each were first generated and r was computed for each of these samples using eq. 1. In each sample, the local optima were then found using DHC for each of the 1,000 points in the sample, and the fitness distance correlation r_l was computed for the local optima. The mean estimates r and r_l over the 100 samples were returned if these mean estimates were within 1% of their true value with 99% probability (assuming Gaussian distribution of the means). If the estimates were not sufficiently accurate, additional 1,000 samples were generated for each of the 100 samples and the procedure was repeated for the increased sample size. Once the number of points in each sample exceeded 10^6 without reaching sufficient accuracy, the procedure was terminated even if the error would not decrease below the threshold; in practice, this case happened only rarely and only for the largest problem sizes.

The correlation length was estimated in a similar way, starting with 100 random walks of 1000 steps each. These walks were used to estimate the correlation length and the autocorrelation coefficient. If both these values were within 1% of their actual value with 99% probability, the mean estimates were returned. Otherwise, the random walks were extended by 1000 points each and the procedure was repeated. The maximum length of each random walk was restricted to 10^6 steps.

The estimate of the escape rate was computed in a simpler manner, since for each value of n, $n/2+1$ estimates must be computed and using more than 100 samples of 1000 points each would become intractable due to the large number of instances considered here. However, even in this case, deviations of the results were rather small and the results thus appeared very accurate.

The distances of local optima from their closest global optimum and the number of steps until the closest local optimum were estimated analogically to the fitness distance correlation and the correlation length.

5.4 Results

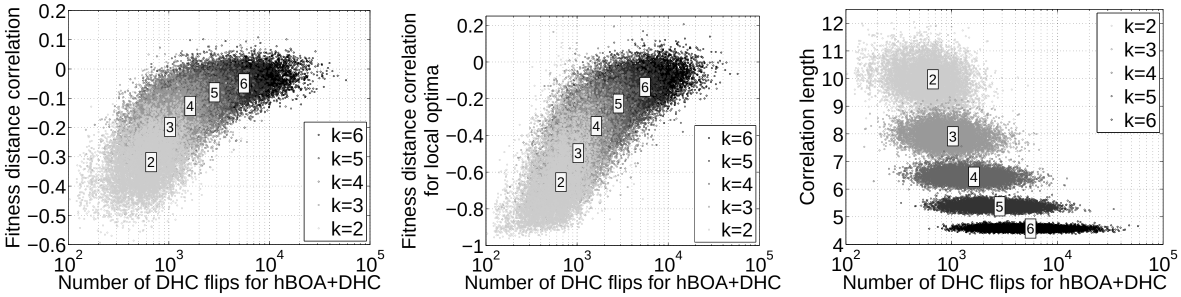
This section presents and discusses the results.

5.4.1 Effects of n and k

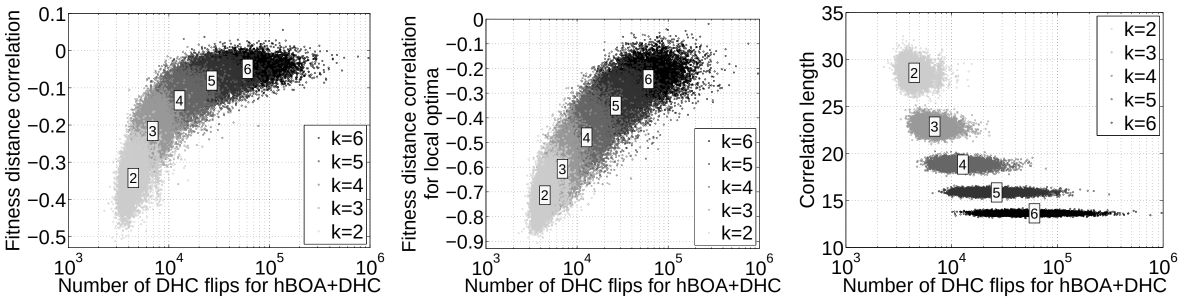
It was empirically shown that for NK landscapes and other additively decomposable problems, problem instances typically become more difficult to solve as the problem size n and the size of subproblems k grow [23]. This behavior was theoretically explained for separable problems for both genetic algorithms as well as estimation of distribution algorithms [8, 11, 18]. The results of the experiments presented in this paper confirm this behavior.

The effects of problem size n on the performance of hBOA+DHC and the measures of problem difficulty are summarized in table 1. Only the results for $k=2$ and $k=6$ are shown; the results for other values of k are qualitatively similar. In all cases, the computational requirements of hBOA+DHC grow with the value of n. Furthermore, the fact that the problem difficulty increases with problem size is also clearly reflected by the correlation length and the distance of local and global optima. However, the influence of problem size on the fitness distance correlation for local optima is nearly negligible, and in some cases the fitness distance correlation for arbitrary candidate solutions is even misleading. The fitness distance correlation thus does not clearly reflect the fact that the instances become more difficult as the problem size increases.

The effects of k on performance of hBOA+DHC and the problem difficulty measures are illustrated in figures 1 and 2. Only the results for several problem sizes are shown; the results for other problem sizes are qualitatively similar. The results confirm that as k grows, the computational requirements of hBOA+DHC grow as well. Furthermore, the effects of k on problem difficulty are well captured by the fitness distance correlation, the correlation length, and the distance of local and global optima.

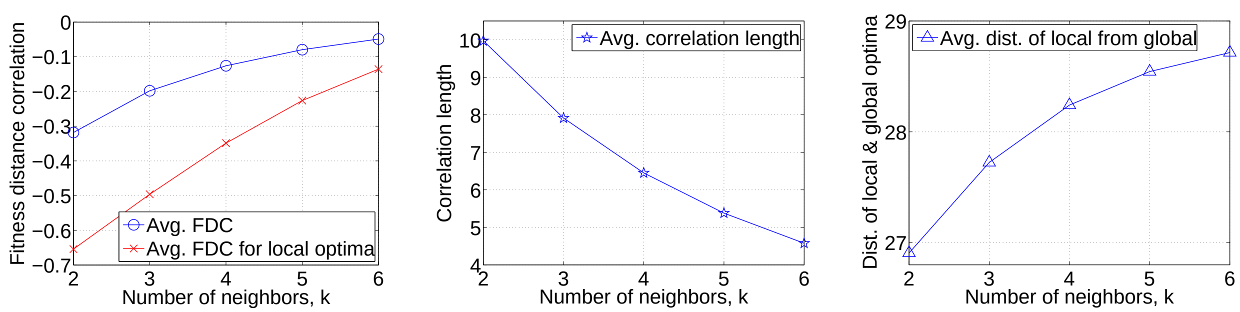


(a) Standard NK landscapes, $n = 36$.

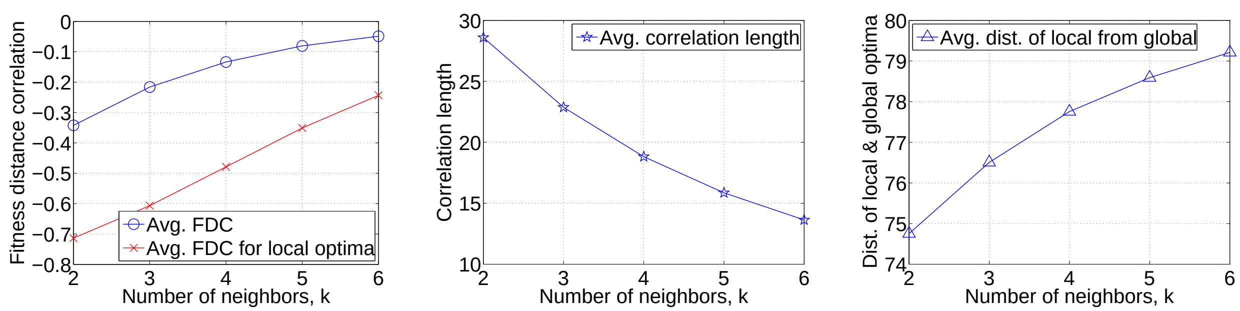


(b) Nearest-neighbor NK landscapes, $n = 100$.

Figure 1: Scatter plot for the fitness distance correlation and the correlation length with respect to the number of DHC flips required by hBOA+DHC to reach the optimum. The results are shown for standard NK landscapes of $n = 36$ bits (top), and nearest-neighbor NK landscapes of $n = 100$ bits (bottom). Boxes with a number k indicate mean values for samples with that k.



(a) Standard NK landscapes, $n = 36$.



(b) Nearest-neighbor NK landscapes, $n = 100$.

Figure 2: The effects of k on the fitness distance correlation, the correlation length, and the average distance of local optima from the global one.

Table 1: Some of the measures of problem difficulty for instances of varying problem size. Standard deviations are shown in brackets.

(a) Standard NK landscapes for $k = 2$ and $k = 6$.

n	k	DHC steps until opt. (hBOA+DHC)	fitness dist. corr.	fitness dist. corr. for local optima	corr. length	avg. dist. of local from global
20	2	147.8 (88.6)	−0.3202 (0.105)	−0.6862 (0.18)	5.30 (0.41)	14.92 (0.63)
28	2	355.9 (195.9)	−0.3177 (0.090)	−0.6624 (0.17)	7.63 (0.50)	20.91 (0.74)
36	2	666.3 (340.2)	−0.3181 (0.079)	−0.6542 (0.16)	9.97 (0.56)	26.91 (0.83)
44	2	1105.0 (515.4)	−0.3145 (0.072)	−0.6424 (0.15)	12.29 (0.61)	32.88 (0.91)
52	2	1633.6 (727.7)	−0.3145 (0.068)	−0.6384 (0.14)	14.62 (0.67)	38.88 (0.98)
20	6	563.2 (308.9)	−0.0496 (0.050)	−0.2086 (0.11)	2.30 (0.05)	15.83 (0.30)
28	6	1916.5 (1148.3)	−0.0491 (0.043)	−0.1605 (0.09)	3.44 (0.06)	22.28 (0.32)
36	6	5583.8 (3940.5)	−0.0489 (0.038)	−0.1355 (0.08)	4.58 (0.07)	28.72 (0.33)

(b) Nearest-neighbor NK landscapes for $k = 2$ and $k = 6$.

n	k	DHC steps until opt. (hBOA+DHC)	fitness dist. corr.	fitness dist. corr. for local optima	corr. length	avg. dist. of local from global
20	2	173.0 (93.0)	−0.3428 (0.104)	−0.7271 (0.151)	5.31 (0.39)	14.96 (0.64)
40	2	873.4 (286.7)	−0.3412 (0.074)	−0.7180 (0.107)	11.13 (0.56)	29.89 (0.91)
60	2	1881.4 (435.9)	−0.3416 (0.061)	−0.7151 (0.087)	16.95 (0.68)	44.85 (1.11)
80	2	3055.0 (691.9)	−0.3416 (0.054)	−0.7143 (0.076)	22.76 (0.80)	59.79 (1.29)
100	2	4436.2 (1019.5)	−0.3423 (0.049)	−0.7134 (0.067)	28.59 (0.88)	74.75 (1.44)
20	6	634.3 (350.1)	−0.0494 (0.051)	−0.2518 (0.122)	2.30 (0.06)	15.80 (0.33)
40	6	5747.2 (3770.7)	−0.0486 (0.036)	−0.2430 (0.091)	5.14 (0.08)	31.68 (0.45)
60	6	16569.5 (10933.3)	−0.0481 (0.030)	−0.2416 (0.075)	7.97 (0.10)	47.53 (0.54)
80	6	34485.9 (23700.1)	−0.0488 (0.026)	−0.2419 (0.065)	10.80 (0.11)	63.37 (0.62)
100	6	60774.8 (42442.8)	−0.0486 (0.024)	−0.2431 (0.058)	13.63 (0.13)	79.21 (0.69)

5.4.2 Measures and actual performance

The previous subsection focused on the influence of n and k on the performance of hBOA+DHC and the measures of problem difficulty. But what is the relationship between the measures of problem difficulty and the actual performance of hBOA+DHC for fixed values of n and k? Furthermore, it is well known that for NK landscapes and other difficult classes of additively decomposable problems, performance of metaheuristics varies substantially from one problem instance to another. Will these variations be captured by the considered measures of problem difficulty?

The relationship between the measures of problem difficulty and the actual computational requirements of hBOA+DHC is illustrated in tables 2 and 3. Only the results for several values of n and k are shown; the results for other values of n and k were qualitatively similar. The results show that the correlation length and both versions of the fitness distance correlation are all in agreement with the actual problem difficulty measured by the number of DHC flips until optimum. However, the distance of local and global optima is misleading in that for the easier problem instances the local optima are in fact further from the global ones than for the more difficult instances. This result is quite surprising. It is also of note that the measures of problem difficulty vary only a little across the instances for the same value of n and k.

5.4.3 Standard vs. nearest-neighbor instances

Table 4 summarizes the results for standard and nearest-neighbor NK landscapes for $n = 36$, and $k = 2$ to $k = 6$. The table compares the actual performance of hBOA+DHC as well as the measures of problem difficulty. The fixed problem size was chosen to focus only on the effects of

Table 2: Some of the measures of problem difficulty for instances of varying difficulty. Standard NK landscapes are considered with $n = 36$, and $k = 2$ or $k = 6$. The difficulty of each instance is measured by the number of DHC steps until optimum using hBOA+DHC. Standard deviations are shown in brackets.

(a) Standard NK landscapes, $n = 36$, $k = 2$.

desc. of instances	DHC steps until opt. (hBOA+DHC)	fitn. dist. corr.	fitn. dist. corr. for local optima	corr. length	avg. dist. of local from global
10% easiest	239.0 (54.6)	−0.3691 (0.074)	−0.7828 (0.112)	10.2259 (0.567)	27.3003 (0.847)
25% easiest	317.9 (78.6)	−0.3549 (0.074)	−0.7457 (0.127)	10.1340 (0.548)	27.1788 (0.839)
50% easiest	422.4 (123.6)	−0.3411 (0.076)	−0.7081 (0.141)	10.0599 (0.553)	27.0721 (0.835)
all instances	666.3 (340.2)	−0.3181 (0.079)	−0.6542 (0.160)	9.9729 (0.555)	26.9069 (0.831)
50% hardest	910.1 (311.8)	−0.2952 (0.076)	−0.6004 (0.160)	9.8859 (0.543)	26.7417 (0.793)
25% hardest	1119.1 (323.4)	−0.2819 (0.075)	−0.5635 (0.161)	9.8345 (0.547)	26.6541 (0.781)
10% hardest	1391.1 (358.7)	−0.2695 (0.076)	−0.5259 (0.158)	9.8102 (0.570)	26.5672 (0.757)

(b) Standard NK landscapes, $n = 36$, $k = 6$.

desc. of instances	DHC steps until opt. (hBOA+DHC)	fitn. dist. corr.	fitn. dist. corr. for local optima	corr. length	avg. dist. of local from global
10% easiest	1785.8 (315.3)	−0.0737 (0.035)	−0.2113 (0.070)	4.5874 (0.069)	28.9966 (0.370)
25% easiest	2279.8 (488.9)	−0.0675 (0.035)	−0.1909 (0.068)	4.5836 (0.067)	28.8979 (0.356)
50% easiest	3034.4 (882.5)	−0.0604 (0.036)	−0.1682 (0.069)	4.5803 (0.067)	28.8242 (0.340)
all instances	5583.8 (3940.5)	−0.0489 (0.038)	−0.1355 (0.075)	4.5752 (0.067)	28.7155 (0.332)
50% hardest	8133.3 (4156.7)	−0.0374 (0.035)	−0.1029 (0.067)	4.5701 (0.066)	28.6067 (0.285)
25% hardest	10738.5 (4534.7)	−0.0316 (0.036)	−0.0868 (0.066)	4.5699 (0.066)	28.5572 (0.272)
10% hardest	14542.1 (5101.1)	−0.0242 (0.036)	−0.0716 (0.064)	4.5652 (0.066)	28.5074 (0.258)

Table 3: Some of the measures of problem difficulty for instances of varying difficulty. Nearest-neighbor NK landscapes are considered with $n = 100$, and $k = 2$ or $k = 6$. The difficulty of each instance is measured by the number of DHC steps until optimum using hBOA+DHC. Standard deviations are shown in brackets.

(a) Nearest-neighbor NK landscapes, $n = 100$, $k = 2$.

desc. of instances	DHC steps until opt. (hBOA+DHC)	fitn. dist. corr.	fitn. dist. corr. for local optima	corr. length	avg. dist. of local from global
10% easiest	3330.9 (163.9)	−0.3763 (0.045)	−0.7594 (0.054)	28.9495 (0.878)	75.2075 (1.370)
25% easiest	3550.2 (217.0)	−0.3657 (0.045)	−0.7466 (0.058)	28.8338 (0.878)	75.1053 (1.395)
50% easiest	3758.6 (265.2)	−0.3578 (0.046)	−0.7364 (0.058)	28.7432 (0.873)	74.9838 (1.427)
all instances	4436.2 (1019.5)	−0.3423 (0.049)	−0.7134 (0.067)	28.5919 (0.885)	74.7529 (1.440)
50% hardest	5113.8 (1044.2)	−0.3269 (0.047)	−0.6905 (0.067)	28.4405 (0.871)	74.5221 (1.415)
25% hardest	5805.5 (1089.4)	−0.3178 (0.047)	−0.6728 (0.069)	28.3831 (0.854)	74.3572 (1.390)
10% hardest	6767.6 (1152.3)	−0.3115 (0.048)	−0.6632 (0.069)	28.3347 (0.865)	74.3459 (1.372)

(b) Nearest-neighbor NK landscapes, $n = 100$, $k = 6$.

desc. of instances	DHC steps until opt. (hBOA+DHC)	fitn. dist. corr.	fitn. dist. corr. for local optima	corr. length	avg. dist. of local from global
10% easiest	21364.1 (2929.9)	−0.0601 (0.023)	−0.2962 (0.050)	13.6468 (0.135)	79.6971 (0.720)
25% easiest	26787.7 (5261.6)	−0.0564 (0.023)	−0.2778 (0.053)	13.6432 (0.131)	79.5491 (0.706)
50% easiest	34276.6 (8833.1)	−0.0529 (0.023)	−0.2626 (0.055)	13.6395 (0.132)	79.4021 (0.687)
all instances	60774.8 (42442.8)	−0.0486 (0.024)	−0.2431 (0.058)	13.6344 (0.131)	79.2125 (0.690)
50% hardest	87272.9 (46049.2)	−0.0444 (0.023)	−0.2237 (0.055)	13.6294 (0.130)	79.0228 (0.638)
25% hardest	114418.9 (52085.3)	−0.0430 (0.023)	−0.2163 (0.056)	13.6274 (0.131)	78.9531 (0.638)
10% hardest	154912.8 (62794.1)	−0.0418 (0.024)	−0.2107 (0.057)	13.6243 (0.130)	78.9277 (0.633)

Table 4: Comparison of the results for the standard NK landscapes (std) and for the nearest-neighbor NK landscapes (nn). In both cases, $n = 36$ and k varies from 2 to 6. Standard deviations are shown in brackets.

(a) Fitness distance correlation

k	DHC steps		fit. dist. corr.		fit. dist. corr. (local)	
	std	nn	std	nn	std	nn
2	666	697	−0.32 (0.08)	−0.34 (0.08)	−0.65 (0.16)	−0.72 (0.11)
3	1029	983	−0.20 (0.07)	−0.21 (0.07)	−0.50 (0.16)	−0.60 (0.13)
4	1629	1437	−0.13 (0.06)	−0.13 (0.06)	−0.35 (0.13)	−0.48 (0.13)
5	2846	2397	−0.08 (0.05)	−0.08 (0.05)	−0.23 (0.10)	−0.35 (0.11)
6	5584	4255	−0.05 (0.04)	−0.05 (0.04)	−0.14 (0.08)	−0.24 (0.09)

(b) Correlation length, and distance of local and global optima

k	DHC steps		corr. length		dist. of local from global	
	std	nn	std	nn	std	nn
2	666	697	9.973 (0.555)	9.962 (0.527)	13.06 (0.72)	12.58 (0.84)
3	1029	983	7.912 (0.342)	7.911 (0.333)	11.33 (0.41)	10.74 (0.55)
4	1629	1437	6.450 (0.200)	6.446 (0.202)	9.96 (0.24)	9.33 (0.33)
5	2846	2397	5.379 (0.116)	5.379 (0.123)	8.86 (0.14)	8.25 (0.20)
6	5584	4255	4.575 (0.067)	4.576 (0.073)	7.95 (0.10)	7.39 (0.12)

the neighborhood and not on the effects of n and k. We would expect the measures of problem difficulty to indicate that problem instances with nearest neighbors are easier than those with standard neighborhoods. This is indeed the case for the fitness distance correlation restricted to local optima. However, the fitness distance correlation for arbitrary solutions and the correlation length provide nearly no indication that nearest-neighbor NK landscapes are easier than those with standard neighborhoods. Furthermore, the distance of local and global optima indicates that standard neighborhoods are easier to solve, although this is clearly not the case.

5.4.4 Escape rate

The effects of the escape rate for NK landscapes with nearest-neighbor interactions for $n = 100$ and $k = 6$ are presented in figure 3. The results for NK landscapes with standard neighborhoods and those for other values of n and k are similar and are thus omitted. The results clearly indicate that the escape rate increases with the number of steps s. Furthermore, the escape rate increases with k regardless of the value of s. Since the difficulty of problem instances increases with k, this indicates that the greater the escape rate, the more difficult the problem becomes. However, further work is necessary to gain better understanding of the escape rate on the actual problem difficulty.

6 Summary and Conclusions

This paper presented a thorough analysis of several measures of problem difficulty on a large number of problem instances of NK landscapes with standard neighborhoods and those with nearest neighbors. Various values of n and k were considered and for each combination of n and k, 10,000 problem instances were generated and tested. Several measures of difficulty were examined for each problem instance and the measures of problem difficulty were compared to the actual computational requirements of hybrid hBOA.

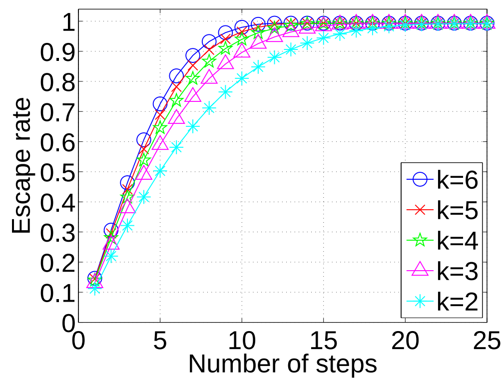


Figure 3: Escape rate for nearest-neighbor NK landscapes of $n = 100$ and $k = 6$.

The paper shows that in most cases the measures of problem difficulty correlate with the actual computational requirements of hybrid hBOA. Most measures capture the effects of n and k on problem difficulty, and they provide input on problem difficulty even for fixed n and k. Nonetheless, in some cases the measures of problem difficulty do not provide a clear indication of what problem instances are difficult and what instances are easy. Furthermore, there is no single measure that would be more accurate than others. Finally, while the differences between the computational requirements of hybrid hBOA across a set of problem instances are substantial, the variance of the problem difficulty measures for this set of instances is typically very low.

One of the interesting topics for future work is to further analyze the results obtained in this paper. Furthermore, analytical approaches may be designed to provide more accurate and faster estimates of the measures of problem difficulty. Additional measures of problem difficulty should also be studied and the relationship between the different measures of problem difficulty should be investigated in more detail. Finally, practical approaches should be proposed to use the measures of problem difficulty for choosing an appropriate optimization method and adequate parameters.

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 Univ. of Missouri in St. Louis through the High Performance Computing Collaboratory sponsored by Information Technology Services, and the Research Award and Research Board programs. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Air Force Office of Scientific Research, or the U.S. Government. Some experiments were done using the hBOA software developed by M. Pelikan and D. E. Goldberg at the Univ. of Illinois at Urbana-Champaign and most experiments were performed on the Beowulf cluster maintained by ITS at the Univ. of Missouri in St. Louis.

References

[1] H. E. Aguirre and K. Tanaka. Genetic algorithms on NK-landscapes: Effects of selection, drift, mutation, and recombination. In G. R. Raidl et al., editors, *Applications of Evol. Comp.: EvoWorkshops 2003*, pages 131–142, 2003.

[2] L. Altenberg. NK landscapes. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages B2.7:5–10. Institute of Physics Publishing and Oxford University Press, 1997.

[3] E. Angel and V. Zissimopoulos. Autocorrelation coefficient for the graph partitioning problem. *Theoretical Computer Science*, 191:229–243, 1998.

[4] 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.

[5] S.-S. Choi, K. Jung, and J. H. Kim. Phase transition in a random NK landscape model. *Genetic and Evolutionary Comp. Conf. (GECCO-2005)*, pages 1241–1248, 2005.

[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] Y. Gao and J. C. Culberson. An analysis of phase transition in NK landscapes. *Journal of Artificial Intelligence Research (JAIR)*, 17:309–332, 2002.

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

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

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

[11] 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.

[12] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 1994.

[13] 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.

[14] 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.

[15] S. Kauffman. *The origins of Order: Self-organization and selection in evolution.* Oxford University Press, 1993.

[16] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–325, 2004.

[17] P. Merz and B. Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):197–213, 2002.

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

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

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

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

[22] M. Pelikan, K. Sastry, M. V. Butz, and D. E. Goldberg. Performance of evolutionary algorithms on random decomposable problems. *Parallel Problem Solving from Nature*, pages 788–797, 2006.

[23] M. Pelikan, K. Sastry, M. V. Butz, and D. E. Goldberg. Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes. *Genetic and Evolutionary Comp. Conf. (GECCO-2008)*, pages 1033–1040, 2008.

[24] 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 Evolutionary Comp. Conf. (GECCO-2009)*, pages 851–858, 2009.

[25] 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.

[26] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[27] A. M. Sutton, L. D. Whitley, and A. E. Howe. A polynomial time computation of the exact correlation structure of k-satisfiability landscapes. *Genetic and Evolutionary Comp. Conf. (GECCO-2009)*, pages 365–372, 2009.

[28] 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.

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

[30] 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.