NK-Fitness Landscapes and Memetic Algorithms with Greedy Operators and k-opt Local Search

Peter Merz

University of Kaiserslautern
Department of Computer Science
P.O. Box 3049, D-67653 Kaiserslautern, Germany
peter.merz@ieee.org

Summary. Memetic algorithms (MAs) with greedy initialization and recombination operators have been successfully applied to several combinatorial optimization problems, including the traveling salesman problem and the graph bipartitioning problem. In this contribution, a k-opt local search heuristic and a greedy heuristic for NK-landscapes are proposed for use in memetic algorithms. The latter is used for the initialization of the population and in a greedy recombination operator. Memetic algorithms with k-opt local search and three different variation operators, including the newly proposed greedy recombination operator, are compared on three types of NK-landscapes. In accordance with the landscape analysis, the MAs with recombination perform better than the MAs with mutation for landscapes with low epistasis. Moreover, the MAs are shown to be superior to previously proposed MAs using 1-opt local search.

1 Introduction

The NK-model of fitness landscapes has been introduced by Kauffman [1, 2] to study gene interaction in biological evolution. In the NK-model, the fitness is the average value of the fitness contributions of the loci in the genome. For each locus, the fitness contribution is a function of the gene value (allele) at the locus and the values of K other interacting genes. Although this is a very simplified model, it allows to produce families of fitness landscapes with interesting properties.

Besides its biological implications, the model is interesting for researchers in the field of evolutionary computation, since the NK-landscape model provides combinatorial optimization problems with tunable difficulty.

In this paper, effective memetic algorithms [3, 4, 5] for NK-landscapes are presented. New greedy and k-opt local search heuristics for NK-landscapes are proposed which can be easily embedded into memetic algorithms. The properties of NK-landscapes are discussed and a fitness distance correlation analysis is performed for the newly introduced heuristic algorithms. It is shown that based on the results of the analysis, the performance algorithms can be predicted: For low epistasis – low values of K in the model – recombination based algorithms are able to exploit the structure of the search space effectively. With increasing epistasis, the landscapes become quickly unstructured, limiting the usefulness of recombination. For high epistasis, mutation based algorithms become favorable over recombination based evolutionary algorithms.

In computer experiments, the effectiveness of sophisticated MAs based on the proposed greedy and k-opt local search heuristics is demonstrated. These algorithms offer (near) optimum solutions in short time even for high dimensional landscapes.

The paper is organized as follows. In section 2, greedy and local search heuristics for the NK-model are introduced. The fitness landscape of three types of NK-models is discussed in section 3. In section 4, results from experiments with memetic algorithms using k-opt local search and three different variation mechanisms are presented. Section 5 concludes the paper and outlines areas of future research.

2 Heuristics for the *NK*-Model

Since *NK*-Landscapes have been studied mainly in the context of simulated biological evolution, little attention has been payed to the development of simple non-evolutionary heuristics. However, besides hill climbing/local search techniques, constructive heuristics such as greedy algorithms can be applied to problems of the *NK*-model.

In the following, a solution vector x is assumed to be a binary vector of length N, i.e. $x = (x_1, \ldots, x_N)$ with the fitness function

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x_i, x_{i_1}, \ldots, x_{i_K}), \tag{1}$$

where the fitness contribution f_i of locus i depends on the value of gene x_i and the values of K other genes $x_{i_1}, \ldots, x_{i_K}$. The function $f_i : \{0, 1\}^{K+1} \rightarrow \mathbb{R}$ assigns a uniformly distributed random number between 0 and 1 to each of its 2^{K+1} inputs. Other random search landscapes have been proposed in [6, 7] which are highly tunable, but will not be investigated in this work.

The NK-model is similar to the unconstrained binary programming problem (BQP) [8]. In fact, the BQP can be regarded as a special case of NK-fitness landscapes with

$$f(x) = \sum_{j=1}^{n} f_i(x) \quad \text{with} \quad f_i(x) = \sum_{j=1}^{n} q_{ij}\, x_i\, x_j, \tag{2}$$

where $Q = (q_{ij})$ is a $n \times n$ matrix. While for NK-landscapes $k(i) = K$ is constant for all i, in the BQP $k(i)$ is defined as the number of non-zero entries in the i-th column of matrix Q. The mean $\overline{k}$ of the $k(i)$ is given by $\overline{k} = n\cdot \text{dens}(Q)$. Due to the strong resemblance of the two problems, heuristics developed for one problem can be applied after small modifications to the other. The heuristics described in the following are similar to the greedy and local search heuristics for the BQP in [9].

2.1 Greedy Algorithms

A point in a *NK*-landscape can be constructed in N steps by assigning in each step a gene value to a gene at a given locus. If the choice of a gene value follows a greedy rule, such an approach can be classified as a greedy heuristic for *NK*-landscapes.

The greedy heuristic proposed in this paper works as follows. A solution is built in N steps by choosing a gene which is still not assigned a value, and a gene value to assign to the gene. The choice is made by maximizing a gain function $g(i,v) : \{1,\ldots,N\} \times \{0,1\} \to \mathbb{R}$ with $g(i,v)$ denoting the gain attained by setting the value of the i-th gene to v. The gain function $g(i,v)$ is defined as the difference between the fitness of a partial solution y with gene i set to v and the fitness of a partial solution x with gene i unspecified: $g(i,v) = f^p(y) - f^p(x)$ with

$$y_j = \begin{cases} v & \text{, if } \; i = j \\ x_j & \text{, otherwise.} \end{cases}$$

The fitness f^p of a partial solution is defined as the average fitness of all solutions matching the template defined by the partial solution: Assume the partial solution x is $x = (1,0,*,0,*,1)$ with $*$ denoting the *don't care* symbol (the gene has no value). Then, the fitness f^p of x is the average fitness of the four solutions $(1,0,\underline{0},0,\underline{0},1)$, $(1,0,\underline{0},0,\underline{1},1)$, $(1,0,\underline{1},0,\underline{0},1)$, and $(1,0,\underline{1},0,\underline{1},1)$.

Assuming the fitness contribution of site i denoted $f_i(x_i, x_{i_1}, \ldots, x_{i_K})$, depends on the site i itself and K neighbors $i_1, \ldots, i_K$, then the neighborhood $N_i = \{i, i_1, \ldots, i_K\}$ defines the set of genes/loci which contribute to the fitness at site i. The set of loci/genes which depend on the value of gene k is thus defined as $D_k = \{i \mid k \in N_i\}$. Hence, the gain function becomes

$$g(i,v) = f^p(y) - f^p(x) = \sum_{i \in D_k} f_i^p(x_i, \ldots, v, \ldots) - f_i^p(x_i, \ldots, x_k, \ldots). \qquad (3)$$

Initially, the partial fitness contribution of locus i is the average over all 2^{K+1} possible values of f_i. Hence, the greedy heuristic based on partial fitness calculations requires more than $n \cdot 2^{K+1}$ additions and is therefore only practically useful for landscapes with small values of K. On the other hand, with increasing K, the solutions produced by the greedy heuristic approach the average fitness of the points in the landscape since for high epistasis the values of f_i^p differ significantly from the values of f_i in the final solution.

The greedy heuristic is randomized by (1) choosing a small fraction ($N/20$) of the genes randomly, and (2) by selecting randomly with a probability proportional to the gains from $\{\arg\max_i g(i,0), \arg\max_i g(i,1)\}$.

2.2 Local Search

The application of local search techniques to *NK*-landscapes is straightforward: Neighboring solutions can be reached by flipping one or more bits simultaneously in the genome. However, instead of calculating the fitness for each neighboring solution anew, it is more efficient to calculate the gain achieved by moving to the new solution. In this context the gain is referred to as the fitness difference between the new and the old solution.

The gain associated with the flipping of a single gene k in the genome x leading to a solution y with

$$y_i = \begin{cases} 1 - x_i & \text{, if} \quad i = k \\ x_i & \text{, otherwise} \end{cases}$$

is the fitness difference of the new solution y and the old solution x:

$$g_k(x) = f(y) - f(x) = \sum_{i \in D_k} f_i(x_i, \ldots, 1 - x_k, \ldots) - f_i(x_i, \ldots, x_k, \ldots). \quad (4)$$

A local search for the *NK*-model can be implemented by maintaining a gain vector $g = (g_1 \ldots, g_N)$ instead of calculating all gains anew in each iteration. After flipping gene k, generally not all of the gains have to be updated. A gain g_i only changes if there is a $j \in D_i$ with $k \in N_j$ or in words the gain of flipping gene i changes if there is a fitness distribution function that depends on the value of gene k and i.

1-opt Local Search

A simple local search based on a 1-opt neighborhood can be realized straightforwardly. The neighborhood is searched by flipping a single bit in the current solution. The gain vector can now be used to find an improving flip in reduced computation time. However, after flipping the gene value, some elements of the gain vector have to be updated accordingly.

Variable k-opt Local Search

The basic scheme described above can be extended to derive more powerful local search algorithms. For example, a *2-opt* local search can be realized by flipping two genes to reach a solution in the neighborhood of the current solution. More generally, a k-opt local search can be realized by flipping k genes simultaneously. Since the neighborhood size of a k-opt local search grows exponentially with k, mechanisms are required to perform a k-opt local search in reasonable time. This can be achieved be considering a small fraction of the k-opt neighborhood similarly to the heuristics by Lin and Kernighan for the TSP [10] and the GBP [11]. The k-opt local search for *NK*-landscapes proposed here is based on the ideas of Lin and Kernighan: in each iteration, a variable number of genes is flipped, depending on a gain criterion. To find the most profitable k-opt move, a sequence of up to n solutions is generated by stepwise flipping genes with the highest associated gain. Every gene is flipped no more than once to guarantee that all solutions in the sequence are different. The solution in the sequence with the highest gain is accepted as the new current solution. This solution may differ in 1 up to n genes depending on the position in the sequence. The pseudo code for the approach is provided in Figure 1. To reduce the running time of the algorithm, the value for the maximum k can be bound to a value smaller than N. Furthermore, the inner repeat loop may be terminated if there was no new x_{best} for more than m solutions.

```
procedure Local-Search-k-opt($x \in X$): $X$;
    begin
        calculate gains $g_i$ for all $i$ in $\{1, \ldots, N\}$;
        repeat
            $x_{prev} := x$, $G_{max} := 0$, $G := 0$, steps = 0, $C := \{1, \ldots, N\}$;
            repeat
                find $j$ with $g_j = \max_{i \in C} g_i$;
                $G := G + g_j$;
                $x_j := 1 - x_j$;
                if $G > G_{max}$ then
                    $G_{max} := G$;
                    $x_{best} := x$;
                endif
                update gains $g_i$ for all $i$;
                $C := C \setminus \{j\}$;
                $steps := steps + 1$;
            until $steps > maxsteps$ or $C = \emptyset$;
            if $G_{max} > 0$ then
                $x := x_{best}$;
            else
                $x := x_{prev}$;
            endif
        until $G_{max} \leq 0$;
        return $x$;
    end;
```

Fig. 1. k-opt Local Search for *NK* Landscapes

3 The Fitness Landscape of the *NK*-Model

The *NK*-model of Kauffman [2, 12] defines a family of fitness landscapes which can be tuned by two parameters: N and K. While N determines the dimension of the search space, K specifies the degree of epistatic interactions of the genes constituting a genome. Each point in the fitness landscape is represented by a bit string of length N and can be viewed as a vertex in the N-dimensional hypercube.

With this model, the "ruggedness" of a fitness landscape can be tuned by changing the value of K and thus the number of interacting genes per locus. Low values of K indicate low epistasis and high values of K represent high epistasis. The two extremes are considered in more detail in the following.

Properties of $K = 0$ Landscapes

The $K = 0$ landscapes have the following properties [2]:

- There is only one 1-opt local/global optimum
- The landscape is smooth; neighboring points (*1-opt* neighbors) in the search space are highly correlated. The fitness of 1-opt neighbors can differ by no more than $\frac{1}{N}$.
- The number of fitter neighbors decreases by one in each iteration of a 1-opt local search.
- The average number of iterations to reach the optimum is $\frac{N}{2}$ and thus in $O(N)$.

For the highest value of K, the properties of the fitness landscapes become quite different.

Properties of $K = N - 1$ Landscapes

If $K = N - 1$, the fitness contribution of a gene depends on the values of all other genes, which results in a highly uncorrelated, rugged fitness landscape. These landscapes have the following properties [2]:

- The expected number of 1-opt local optima is $\frac{2^N}{N+1}$
- The expected fraction of fitter 1-opt neighbors dwindles by $\frac{1}{2}$ after each iteration of a 1-opt local search
- The expected number of improvement steps to reach a 1-opt local optimum is in $O(\log N)$
- The expected number of solutions to examine for reaching a 1-opt local optimum is proportional to N
- The ratio of accepted to tried moves scales as $\log N/N$
- Starting from an arbitrary solution, only a small fraction of local optima ($\leq N^{\log_2(N-1)/2}$) can be reached by a 1-opt local search.

- Only from a small fraction of starting solutions ($2^{(\log_2 N)^2/2}$), the global optimum can be reached by a 1-opt local search.

Furthermore, Kauffman [2] has shown that for increasing N, the fitness values of the local optima decrease towards $\frac{1}{2}$. He calls this phenomenon *a complexity catastrophe.*

Random vs. Adjacent Neighbor Model

Besides the values for the parameters N and K, the choice of the neighbor model is important for *NK*-landscapes, too. Kauffman [2] distinguishes two variants, the *random neighbor model* and the *adjacent neighbor model.* In the former, the genes which contribute to the fitness at locus i are chosen at random. In other words, the neighbors i_1 through i_K are randomly selected among the N. In the latter, the i_1 through i_k are the nearest loci to the gene at locus i.

The landscape properties described above are independent of the neighbor model. However, Weinberger [13] has shown that the computational complexity of both models differs. He was able to show that the *NK* decision problem with adjacent neighbors is solvable in $O(2^K N)$ steps and is thus in $\mathcal{P}$ and that the *NK* decision problem with random neighbors is $\mathcal{NP}$-complete for $K \geq 3$.

3.1 Autocorrelation Analysis

To measure of the ruggedness of a fitness landscape, Weinberger [14] suggests the use of (auto)correlation functions. The *autocorrelation function* $\rho(d)$ [15, 14] reflects the fitness correlation of points at distance d in the search space. Weinberger [16] derived formulas for the autocorrelation function of *NK*-landscapes. He found that the autocorrelation function $\rho(d)$ depends on the neighbor model of the landscape. In the random neighbor model, the autocorrelation function becomes

$$\rho(d) = \left(1 - \frac{d}{N}\right)\left(1 - \frac{K}{N-1}\right)^d, \tag{5}$$

and for the adjacent neighbor model, ρ becomes

$$\rho(d) = 1 - \frac{d(K+1)}{N} + \frac{d}{N\binom{N-1}{d-1}} \sum_{l=1}^{K} (K+1-l)\binom{N-l-1}{d-2}, \tag{6}$$

with d denoting the hamming distance between bit vectors.

Alternatively, Weinberger suggested to perform random walks to investigate the correlation structure of a landscape. The *random walk correlation function* $r(s)$ [14, 17, 18] of a time series $\{f(x_t)\}$ defines the correlation of two points s steps away along a random walk through the fitness landscape.

The random walk correlation function for the *NK*-model has been calculated by Fontana *et al.* [19]:

$$r(s) \approx \left(1 - \frac{K+1}{N}\right)^s \tag{7}$$

for the adjacent and random neighbor model.

If the time series is ***isotropic***, ***Gaussian*** and ***Markovian*** [14], then the corresponding landscape is called AR(1) landscape, and the random walk correlation function is of the form $r(s) = r(1)^s = e^{-s/\ell}$ with ℓ being the correlation length of the landscape. Hence, the correlation length ℓ [18] of the landscape is defined as

$$\ell = -\frac{1}{\ln(|r(1)|)} = -\frac{1}{\ln(|\rho(1)|)} \tag{8}$$

for $r(1), \rho(1) \neq 0$. The correlation length directly reflects the ruggedness of a landscape: the lower the value for ℓ, the more rugged the landscape. In the NK-model, the correlation length is for adjacent and random neighbors

$$\ell \approx \frac{N}{K+1}. \tag{9}$$

It is not surprising that the correlation length decreases with increasing K.

The formula show that the *NK*-model allows to produce landscapes with arbitrary ruggedness. The correlation length can be set to 1 by choosing $K = N - 1$ leading to a totally random landscape with uncorrelated neighboring points. Choosing the other extreme $K = 0$, the correlation length grows to its maximum value: N, resulting in a smooth, single peaked landscape.

3.2 Fitness Distance Correlation Analysis

The fitness distance correlation (FDC) coefficient is known to be an important measure in the context of fitness landscapes, proposed in [20] as a measure for problem difficulty for genetic algorithms. The FDC coefficient ϱ is defined as

$$\varrho(f, d_{\mathrm{opt}}) = \frac{\mathrm{Cov}(f, d_{\mathrm{opt}})}{\sigma(f)\,\sigma(d_{\mathrm{opt}})} \approx \frac{1}{\sigma(f)\sigma(d)}\,\frac{1}{m}\sum_{i=1}^{m}(f_i - \overline{f})(d_i - \overline{d}), \tag{10}$$

given a set of points $x_1, x_2, \ldots, x_m$ with $f_i = f(x_i)$ denoting the objective value, $d_i = d_{opt}(x_i)$ denoting the shortest distance to a global optimum solution, and $\sigma(f)$ and $\sigma(d)$ denoting the standard deviation of f and d, respectively.

In his studies of *NK*-landscapes, Kauffman [2] investigated the correlation of fitness and distance to the optimum of local optimum solutions with respect to 1-opt local search. In this work, the analysis is extended by investigating fitness distance correlation with respect to the greedy heuristic and k-opt local search. Experiments were conducted for three selected instances with N fixed to 1024, K in $\{2, 4, 11\}$ and a random neighbor model. Since the optimum solutions for these instances are not known, the best solutions found with the MAs described below in long runs (14400 s on a Pentium II 300 MHz PC) are used instead. These solutions are likely to be the global optima or at least close to the global optima with respect to fitness and distance.

In the first experiment, the distribution of greedy solutions in the search space is investigated. The results of the analysis are summarized in Table 1. In the first column, the name of the instance is displayed, and in the second and third column the parameters N and K are given. In columns four through eight, the minimum distance of the greedy solutions to the expected global optimum ($\min d_{opt}$), the average distance of greedy solutions to the global optimum ($\overline{d}_{opt}$), the average distance between the greedy solutions ($\overline{d}_{gr}$), the number of distinct greedy solutions (N_{gr}) out of 2500, and the fitness distance correlation coefficient (ϱ) are provided, respectively. Additionally, the normalized average distance, i.e. the average distance of the local optima to the global optimum divided by the maximum distance in the search space N is shown in column five in parentheses.

Table 1. Results of the Fitness Distance Analysis of Greedy Solutions.

Instance	N	K	$\min d_{opt}$	$\overline{d}_{opt}$	$\overline{d}_{gr}$	N_{gr}	ϱ
C2-1024	1024	2	130	220.62 (0.22)	195.03	2500	-0.62
D4-1024	1024	4	264	372.29 (0.36)	377.38	2500	-0.24
B11-1024	1024	11	458	515.74 (0.50)	469.35	2500	-0.01

For small K, the greedy solutions are close to each other and close to the best known solution. There is a correlation between fitness and distance to the best known solution as the value ρ indicates. About three quarters of the gene values are equal in all greedy solutions for $K = 2$ and thus the solutions are contained in a small fraction of the search space. With increasing K, average distance between the greedy solutions quickly converges to the average distance ($N/2$) of the solutions in the search space. Surprisingly, already at $K = 11$ there is no correlation between greedy solutions and they have random distribution in the search space as expected for large values of K.

In the second experiment, the correlation of fitness and distance to the best known solution of k-opt solutions was investigated. The results are shown in Table 2. Again, in the first column, the name of the instance is displayed, and in the second and third column the parameters N and K are given. In columns four through eight, the minimum distance of the locally optimal solutions to the expected global optimum ($\min d_{opt}$), the average distance of the local optima to the global optimum ($\overline{d}_{opt}$), the average distance between the local optima ($\overline{d}_{loc}$), the number of distinct local optima (N_{k-opt}) out of 2500, and the fitness distance correlation coefficient (ϱ) are provided, respectively. Additionally, the normalized average distance, i.e. the average distance of the

Table 2. Fitness Distance Correlation Analysis of k-opt Solutions.

Instance	N	K	$\min d_{opt}$	$\overline{d}_{opt}$	$\overline{d}_{loc}$	N_{k-opt}	ϱ
C2-1024	1024	2	191	301.47 (0.29)	346.16	2500	-0.65
D4-1024	1024	4	347	440.57 (0.43)	470.36	2500	-0.33
B11-1024	1024	11	459	511.88 (0.50)	511.72	2500	0.02

local optima to the global optimum divided by the maximum distance in the search space N is shown in column five in parentheses. Similarly as for the greedy heuristic, the average distance between the local optima and the average distance to the best known solution increases quickly with increasing K. At $K = 11$ there is no correlation between fitness and distance, and the distribution is similar to a uniform distribution of random points in the search space. There is slightly higher correlation in case of k-opt in comparison to 1-opt in case of the $K = 2, 4$ landscapes. However, greedy solutions have even a shorter minimum and average distance to the best known solution than k-opt solutions. In addition to Tables 1 and 2, fitness distance plots for the three instances are shown in Figure 2. On the left, the scatter plots for 2500 greedy solutions are provided, and on the right the scatter plots for 2500 k-opt solutions are displayed. For $K = 2$, the orientation of the points towards the origin is obvious. The cloud of points "moves" with increasing K quickly to the middle of the plane losing the orientation to the origin and thus to the optimum. These results correspond to the findings of Kauffman [2] for 1-opt local search. He further observed that for instances of the adjacent neighbor model the correlation of fitness and distances decreases not as rapidly as for the random neighbor model with increasing K.

From the perspective of performance prediction of MAs, the analysis provides some useful information. For small K (< 5) , recombination-based memetic algorithms are expected to have a good performance since with recombination the fitness distance correlation of the local optima can be exploited: With increasing fitness, the local optima are closer together, and their distance to the optimum becomes smaller. Furthermore, the locally optimal solutions are found in a small region of the search space in which the global optimum has a more or less central position. The greedy heuristic is very well suited for these instances with low epistasis and it is therefore promising to include the heuristic in the initialization phase of the population as well as in the recombination step. For larger K, the effectiveness of recombination decreases and eventually mutation based MAs are better suited.

3.3 Alternative Distance Measures

The fitness distance correlation analysis requires a feasible distance measure for the search space. In case of bit-strings, the hamming distance appears to

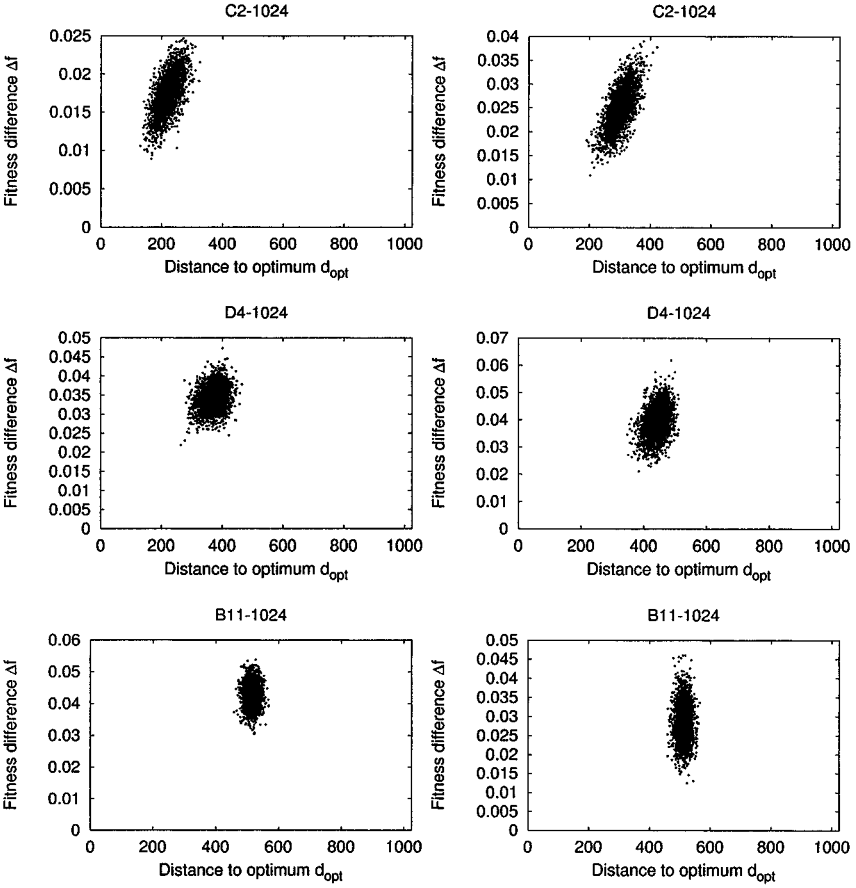

Fig. 2. Fitness-Distance Plots of Greedy Solutions (left) and *k*-opt Solutions (right)

be a natural choice. However, the hamming distance does not reflect exactly how a *k*-opt local search "sees" the landscape. Alternatively, an edit distance may be considered which counts the changes required for a *k*-opt local search to convert one solution to the other. However, a problem arises with such an approach, since the *k*-opt local search is not capable of converting all solutions into all other solutions. Besides the fact that only better solutions are produced by a *k*-opt, per definition not all better solutions are found by the local search. The hamming distance is a lower bound of the number of steps (flips) required for a *k*-opt local search to convert a solution to another assuming that it can. Essentially, a 1-opt local search and a *k*-opt local search are based on single flips, only the acceptance criterion is different in *k*-opt. Note, that the *k*-opt local search discussed in this paper considers only a sequence of order one flips, not, for example, all pairs of order two flips (as would be in a true 2-opt local search). Finally, the FDC analysis may provide hints how the evolutionary part of a MA "sees" the landscape. Here, the hamming distance appears still to be a suitable choice, since properties like respectfulness and assortedness [21, 22] can be described with this distance measure.

4 Memetic Algorithms for *NK* Landscapes

Memetic algorithms have been applied with great success to several combinatorial optimization problems. In this paper, we focus on a class of memetic algorithms that uses a simple evolutionary framework with a single panmictic population instead of spatially structured populations [23], or tree-structured populations [24]. Furthermore, we concentrate on using a single local search strategy in contrast to the self-adaptation of the local search strategy [25]. The framework is thus rather simple and derived from other evolutionary algorithms, with the only difference that after initialization and after recombination or mutation, a local search procedure is applied to assure that all individuals in the population are local optima. This simple framework has been successfully used in studies for several combinatorial problems, including the graph bipartitioning problem [26], the quadratic assignment problem [27], the traveling salesman problem [28], and the binary quadratic programming problem [29].

The application of MAs to *NK*-landscapes is straightforward. Since problems of the *NK*-model are binary-coded, all GA variation operators such as k-point crossover and bit-flip mutation for bit strings can be used in a MA. As shown in [30], genetic algorithms do not scale well with problem size N. They perform much worse than memetic algorithms for a problem size $N \geq 512$. Therefore, we concentrate in the following on the hardest landscapes from the studies in [30] with $N = 1024$ and varying K.

4.1 Population Initialization and Local Search

The population can be initialized by randomly generating bit strings and by subsequently applying local search. For low values of K, the use of the randomized greedy heuristic described above can be used alternatively in combination with local search. Suitable local search algorithms are 1-opt local search and k-opt local search as described above.

4.2 Evolutionary Variation Operators

Due to the binary coding of the problem, all operators on binary strings can be applied in an evolutionary algorithm and therefore in a memetic algorithm, such as single point or two-point crossover, uniform crossover and bit flip mutation operators.

Recombination

A variant of uniform crossover (UX) that is used in the CHC algorithm of Eshelman [31] is an alternative to the crossover operators noted above. The operator creates (with high probability) offspring that have a maximum Hamming distance to the parents which is half of the distance between the parents themselves. The operator is called denoted *HUX* in the following.

Alternatively, the greedy construction scheme can be used in recombination to produce offspring. A *greedy recombination operator* denoted *GX* is therefore devised that works by first inheriting all the gene values that are common to the two parents to retain respectful recombination [22]. Then the remaining loci are set making greedy choices as in the greedy heuristic described above. This operator is especially effective for problems with low epistasis.

Mutation

Simple bit flip mutation is not useful in a memetic algorithm, since the flipping of a single bit will be reversed by a subsequently performed local search with a high probability. Hence more than one bit must be flipped simultaneously in the parent solution. If p bits are flipped by the mutation operator, the Hamming distance of the resulting offspring and the original parent solution becomes p. The value of p should be chosen to minimize the probability that the subsequent local search rediscovers the unmutated solution.

4.3 Selection and Restarts

Selection for reproduction is performed on a purely random basis without bias to fitter individuals, while selection for survival is achieved by choosing the best individuals from the pool of parents and children. Thus, replacement in our algorithm is similar to the selection in the $(\mu + \lambda)$-ES [32]. Additionally, duplicates will be replaced by other solutions, so that each phenotype exists only once in the new population.

In order to circumvent the problem of premature convergence, cataclysmic mutations [31] are performed when the population has converged. The mutation operator is applied to all but the best individual in the population, where p is determined by a third of the average Hamming distance between the individuals in the initial population. This value for p exhibited good performance in several experiments.

5 Performance Evaluation

We studied the performance of the memetic algorithms described above in several experiments. The results are discussed in the following starting with an evaluation of the components, namely the greedy and local search heuristics. All experiments were performed on a Pentium II PC (300 MHz). The algorithms were implemented in C++.

5.1 Variable k-opt Local Search Variants

Running time and solution quality of the k-opt local search highly depend on the termination criterion of the inner loop, in other words, the maximum number of steps (search depth) considered in each iteration.

In order to investigate the influence of the search depth termination criterion we tested three variants of the local search procedure in Fig. 1. The full k-opt variant is exactly as shown in the figure, with *maxsteps* set to N. In the fast variant, the inner loop is terminated if there was no new x_{best} for more than $m = 40$ steps and the number of *maxsteps* was set to $N/2$. Finally, a simple tabu search variant was considered. In this variant the inner loop is terminated as soon as a better solution has been found. It is essentially a tabu search with a memory of N solutions and no aspiration criterion. The results of the comparison is displayed in Table 3. In the table, the percentage deviation from the best known solution as well as the relative performance in respect to the fast variant are provided (larger values denote higher run times). As the figures suggest, the full variant is approximately 3 up to 5 times slower than the fast variant with only slightly better average objective values. Hence, the extra running time for the full variant appears not to be justified. The tabu search variant is much slower (up to 20 times) than the fast variant and also clearly inferior in average solution quality. Therefore, the fast variant is used in all remaining experiments.

Table 3. Comparison of k-opt Local Search Variants

Instance	Fast k-opt		Full k-opt		Tabu k-opt	
C2-1024	3.322%	1.0	3.263%	3.1	3.708%	20.4
D4-1024	4.918%	1.0	4.810%	2.7	5.614%	10.3
B11-1024	3.697%	1.0	3.571%	2.7	4.427%	2.5

An interesting issue is how the dynamics of a k-opt local search change if the landscapes become more rugged: The number of K and the search depth of the k-opt local search may be related. To investigate this issue the local search variants were compared in respect to the average number of steps per iteration and the number of iterations required to find a local optimum. The findings are summarized in Fig. 3 and Fig. 4. In the left plot of Fig. 3, the average number of flips performed in each iteration of the fast variant are displayed. As can be seen, the number of flips is initially very high and slightly less than $N/2$ for $K = 2$. Not surprisingly, the number is much lower for $K = 11$ due to the rapidly decreasing (auto-)correlation function of the landscape. As

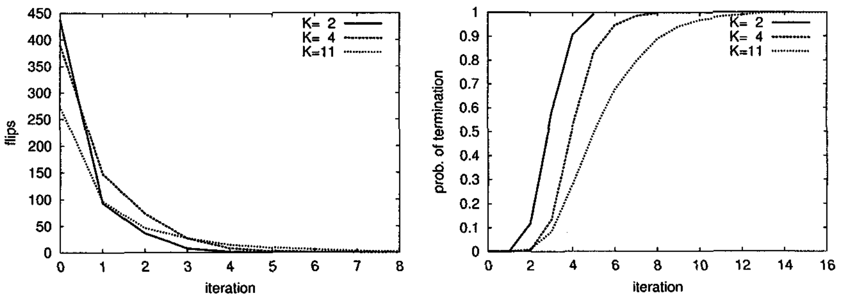

Fig. 3. k-opt Local Search Statistics for the Fast k-opt Variant

shown in the right of the figure, the number of iterations to reach a local optimum is very low (below 14 iterations), and increases with K. In the tabu search variant the expected number of iterations is much higher since flips are performed immediately, when an improving flip is found. The probability of termination is provided in the right plot of Fig. 4. Up to 800 iterations are required for $N = 1024$. The plot in the left hand side of the figure shows the frequency of k-flips depending on k on a logarithmic scale. Again, the frequencies of the $K = 11$ landscape are lower than those of the other two landscapes with $K = 2$ and $K = 4$, and the frequencies decrease exponentially. These results indicate that the optimum number of k (the search depth) in a k-opt local search should be dynamically chosen and not to be fixed in advance. In MAs where this parameter is adapted, it should be ensured that the parameter can be adjusted fast enough to meet the requirements at the current state of the search.

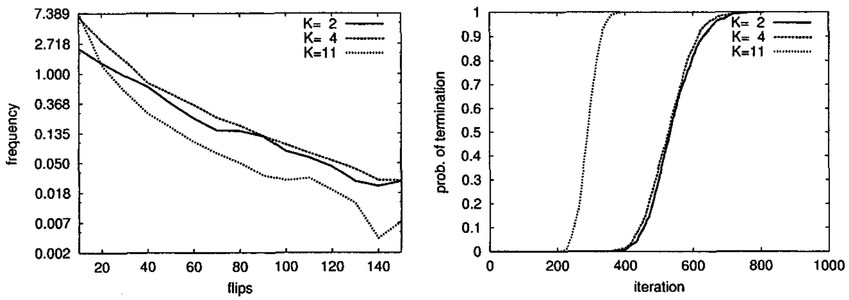

Fig. 4. k-opt Local Search Statistics for the Tabu k-opt Variant

5.2 Greedy and Local Search

To investigate the relative performance of the greedy heuristic and the k-opt local search, experiments were conducted in which the two together with the

1-opt local search were applied to the three landscapes with $N = 1024$ used in the analysis above. The results are shown in Table 4. In the table, the average

Table 4. Performance of the Greedy Heuristic, 1-opt and k-opt Local Search.

	Greedy		**1-opt LS**		**k-opt LS**	
Instance	fitness	t/ms	fitness	t/ms	fitness	t/ms
C2-1024	0.7326 (2.33%)	76.9	0.7135 (4.88%)	19.7	0.7251 (3.32%)	52.3
D4-1024	0.7563 (4.34%)	173.2	0.7237 (8.47%)	28.4	0.7515 (4.94%)	114.8
B11-1024	0.7262 (5.56%)	22120	0.7094 (7.74%)	112.5	0.7403 (3.72%)	677.3

performance (fitness and average percentage excess in parentheses) and the average running time (t/ms) in milliseconds of a single run, is shown for the greedy heuristic and 1-opt and k-opt local search applied to randomly generated solutions. The values are averaged over 10000 runs except for the greedy heuristic and the problem instance B11-1024: Due to the long running time, 1000 runs were performed instead of 10000. The values given in parentheses denote the deviation from the best known solution in percent.

For $K = 2$ and $K = 4$, the greedy heuristic outperforms the local searches but requires more CPU time. For $K = 11$, the k-opt local search dominates over the two others. The CPU time required for a run of the greedy algorithm exceeds 22 seconds and is thus more than 32 times higher than for k-opt local search rendering the greedy heuristic impractical for such relative large K. For $K = 2$, the greedy heuristic is furthermore capable of producing comparable results to a GA in a single run and thus in 173 milliseconds, where the GA requires 1200 seconds [30]. Also for $K = 4$ and $K = 11$, the GAs in [30] are outperformed by the greedy heuristic and the k-opt local search in a single run, demonstrating even more drastically the inferior performance of traditional GAs on relatively large instances.

5.3 Memetic Algorithms with k-opt Local Search

To assess the performance of memetic algorithms with k-opt, additional experiments have been conducted. With the same time limit (1200 seconds) as chosen for the comparison of genetic algorithms with MAs in [30], the MAs with k-opt local search were applied to the three instances of size 1024. With a population size of 40, the production of 20 new offspring per generation, and restarts enabled as in [30], the MA were run with three different variation operators. The first MA uses the greedy heuristic in the initialization phase and the greedy recombination operator (GX). The second MA uses HUX as the recombination operator and the third MA uses the mutation operator (MUT) described above with $p = 3$. The results of the experiments are summarized in Table 5. For each algorithm, the average number of generations (gen) pro-

Table 5. Performance of k-opt Local Search MAs with three types of variation.

	C2-1024		D4-1024		B11-1024	
Op	gen	fitness, quality	gen	fitness, quality	gen	fitness, quality
GX	12505	0.750002, 0.01%	5750	0.787570, 0.39%		
HUX	11954	0.750009, 0.01%	5730	0.786874, 0.48%	216	0.753565, 1.99%
MUT	6402	0.744757, 0.71%	4306	0.772776, 2.26%	704	0.755747, 1.71%
HUX1	12615	0.748230, 0.25%	4540	0.783665, 0.89%	105	0.732874, 4.91%
Best		0.750065, 0.00%		0.790640, 0.00%		0.768882, 0.00%

duced is provided as well as the average fitness (fitness) of the final solution along with the percentage excess over the best known solution (quality). The results of the MA with 1-opt local search and HUX recombination (denoted HUX1) from [30] are included for easy comparison. Due to the long running times for the greedy heuristic on B11-1024, the MA with GX was not tested on this landscape.

For $K = 2$, the MA with greedy recombination and HUX recombination perform equally well. Both find the best known solution in one out of 20 runs and have the same worst result. For $K = 4$ and $K = 11$, the greedy recombination MA outperforms the others. The mutation based MA is as expected the worst out of the three for $K = 2$ and $K = 4$. For $K = 11$, the mutation based MA achieves a better average result than the MA with HUX. The same tendency appeared in the results of the MAs with 1-opt local search [30]: for the unstructured landscape with $K = 11$, recombination has no benefit compared to mutation. The recombination based MAs with k-opt local search perform clearly better than the algorithms with 1-opt local search in [30]. In particular new best solutions have been found for the three landscapes. Summarizing, the k-opt MAs have a higher potential and perform better if longer running times are chosen.

6 Conclusions

NK-landscapes have been introduced as a formal model of gene interaction in biological evolution, and since they are random, several statistical properties of the landscapes are known. To derive highly effective memetic algorithms for the *NK*-model, two new heuristics have been proposed, a greedy algorithm and a k-opt local search. The distribution of the solutions produced by these heuristics has been analyzed by performing a fitness distance correlation analysis on selected instances. The results allow to predict when greedy choices based on the greedy heuristic are favorable in a memetic framework and when not. Additionally, investigating the distribution of local optima in the landscapes allows to determine whether or not recombination is effective.

The greedy heuristic incorporated in the initialization phase as well as in the recombination operator of a MA with k-opt local search is shown to be highly effective for landscapes with low epistasis. The landscape analysis has shown that with increasing epistasis, the landscape becomes rapidly unstructured. Thus, for these instances, a k-opt local search MA with mutation instead of recombination has been shown to be favorable.

Moreover, the memetic algorithms with k-opt local search are shown to outperform previously proposed memetic algorithms with 1-opt local search: new best solutions have been found with the former for three landscapes.

There are several issues for future research. Firstly, the algorithms and landscape studies should be extended to cover other random search landscapes [6, 7]. Secondly, random walk correlation analysis may be applied on paths between local optima in the spirit of path relinking [33] to gain more insight in the effectiveness of recombination in memetic algorithm frameworks. Finally, the potentials of the algorithms described in the paper have to be investigated in other application domains of practical interest.

References

1. Kauffman, S.A.: Emergent Properties in Random Complex Automata. Physica D **10** (1984)
2. Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press (1993)
3. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report C3P Report 826, Caltech Concurrent Computation Program, California Institue of Technology (1989)
4. Moscato, P.: Memetic Algorithms: A Short Introduction. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw–Hill, London (1999) 219–234
5. Merz, P.: Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany (2000)
6. Smith, R.E., Smith, J.E.: An Examination of Tuneable, Random Search Landscapes. In Banzhaf, W., Reeves, C., eds.: Foundations of Genetic Algorithms 5. Morgan Kaufmann, San Francisco, CA (1999) 165–181
7. Smith, R.E., Smith, J.E.: New Methods for Tunable, Random Landscapes. In Martin, W.N., Spears, W.M., eds.: Foundations of Genetic Algorithms 6. Morgan Kaufmann, San Francisco (2001) 47–67
8. Laughunn, D.J.: Quadratic Binary Programming. Operations Research **14** (1970) 454–461
9. Merz, P., Freisleben, B.: Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming. Journal of Heuristics **8** (2002) 197–213
10. Lin, S., Kernighan, B.: An Effective Heuristic Algorithm for the Traveling Salesman Problem. Operations Research **21** (1973) 498–516

11. Kernighan, B., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. Bell Systems Journal **49** (1972) 291–307
12. Kauffman, S.A., Levin, S.: Towards a General Theory of Adaptive Walks on Rugged Landscapes. Journal of Theoretical Biology **128** (1987) 11–45
13. Weinberger, E.D.: NP Completeness of Kauffman's N-k Model, A Tuneable Rugged Fitness Landscape. Technical Report 96-02-003, Santa Fe Institute, Santa Fe, New Mexico (1996)
14. Weinberger, E.D.: Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. Biological Cybernetics **63** (1990) 325–336
15. Stadler, P.F.: Correlation in Landscapes of Combinatorial Optimization Problems. Europhysics Letters **20** (1992) 479–482
16. Weinberger, E.D.: Local Properties of Kauffman's N-k model: A Tunably Rugged Energy Landscape. Physical Review A **44** (1991) 6399–6413
17. Stadler, P.F.: Towards a Theory of Landscapes. In Lopéz-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., Zertuche, F., eds.: Complex Systems and Binary Networks. Volume 461 of Lecture Notes in Physics., Berlin, New York, Springer Verlag (1995) 77–163
18. Stadler, P.F.: Landscapes and their Correlation Functions. Joural of Mathematical Chemistry **20** (1996) 1–45
19. Fontana, W., Stadler, P.F., Bornberg-Bauer, E.G., Griesmacher, T., Hofacker, I.L., Tacker, M., Tarazona, P., Weinberger, E.D., Schuster, P.: RNA Folding Landscapes and Combinatory Landscapes. Physcal Review E **47** (1993) 2083–2099
20. Jones, T., Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In Eshelman, L.J., ed.: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann (1995) 184–192
21. Radcliffe, N., Surry, P.: Formal Memetic Algorithms. In Fogarty, T., ed.: Evolutionary Computing: AISB Workshop. Volume 865 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (1994) 1–16
22. Radcliffe, N., Surry, P.: Fitness Variance of Formae and Performance Prediction. In Whitley, L., Vose, M., eds.: Proceedings of the Third Workshop on Foundations of Genetic Algorithms, San Francisco, Morgan Kaufmann (1994) 51–72
23. Gorges-Schleuter, M.: ASPARAGOS: An Asynchronous Parallel Genetic Optimization Strategy. In Schaffer, J.D., ed.: Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann (1989) 422–427
24. Moscato, P., Tinetti, F.: Blending Heuristics with a Population-based Approach: A Memetic Algorithm for the Traveling Salesman Problem. Technical Report CeTAD, CeTAD, Universitad Nacional de La Plata (1994)
25. Krasnogor, N., Smith, J.: Emergence of profitable search strategies based on a simple inheritance mechanism. In Lee Spector *et al.*, ed.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), San Francisco, California, USA, Morgan Kaufmann (2001) 432–439
26. Merz, P., Freisleben, B.: Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. Evolutionary Computation **8** (2000) 61–91
27. Merz, P., Freisleben, B.: Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem. IEEE Transactions on Evolutionary Computation **4** (2000) 337–352

28. Merz, P., Freisleben, B.: Memetic Algorithms for the Traveling Salesman Problem. Complex Systems **13** (2001) 297–345
29. Merz, P., Katayama, K.: Memetic Algorithms for the Unconstrained Binary Quadratic Programming Problem. Bio Systems (2002) To appear.
30. Merz, P., Freisleben, B.: On the Effectiveness of Evolutionary Search in High–Dimensional NK-Landscapes. In Fogel, D., ed.: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Piscataway, NJ, IEEE Press (1998) 741–745
31. Eshelman, L.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In Rawlings, G.J.E., ed.: Foundations of Genetic Algorithms. Morgan Kaufmann (1991) 265–283
32. Schwefel, H.P.: Numerische Optimierung von Computer–Modellen mittels der Evolutionsstrategie. Volume 26 of Interdisciplinary Systems Research. Birkhäuser Verlag, Basel (1977)
33. Glover, F.: Scatter Search and Path Relinking. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, London (1999) 297–316