

Chapter 3

Fitness Landscapes and Memetic Algorithm Design

3.1 Introduction

The notion of fitness landscapes has been introduced to describe the dynamics of evolutionary adaptation in nature [40] and has become a powerful concept in evolutionary theory. Fitness landscapes are equally well suited to describe the behavior of heuristic search methods in optimization, since the process of evolution can be thought of as searching a collection of genotypes in order to find the genotype of an organism with highest fitness and thus highest chance of survival.

Thinking of a heuristic search method as a strategy to “navigate” in the fitness landscape of a given optimization problem may help in predicting the performance of a heuristic search algorithm if the structure of the landscape is known in advance. Furthermore, the analysis of fitness landscapes may help in designing highly effective search algorithms. In the following we show how the analysis of fitness landscapes of combinatorial optimization problems can aid in designing the components of memetic algorithms. However, some of the presented concepts can also be utilized for the development of other search algorithms, including genetic algorithms and neighborhood search algorithms (e.g. simulated annealing and tabu search).

3.2 Fitness Landscapes of Combinatorial Problems

In combinatorial optimization, the number of (candidate) solutions of a given problem is finite. Due to the fact that the complete enumeration of the search space is in many cases impractical (many combinatorial optimization problems are known to be NP-hard [12]), only a small fraction of all solutions can be evaluated and thus the structure of the problem must be exploited to find optimum or near optimum solutions. To identify the structure of a given problem, the idea of fitness landscape analysis appears to be a promising approach.

3.2.1 Combinatorial Optimization Problems

An example of a combinatorial optimization problem addressed in this chapter is the famous *traveling salesman problem* (TSP) in which a salesman tries to find a shortest closed tour to visit a set of N cities under the condition that each city is visited exactly once [19]. Thus, the TSP consists of finding a permutation π of the set $\{1, 2, 3, \dots, n\}$ that minimizes the quantity

$$C(\pi) = \sum_{i=1}^{n-1} d_{\pi(i), \pi(i+1)} + d_{\pi(n), \pi(1)} \quad (3.1)$$

where d_{ij} denotes the distance between city i and j .

Another well-known combinatorial optimization problem is the *graph bi-partitioning problem* (GBP), which can be stated as follows. Given a undirected Graph $G = (V, E)$, the GBP is to find a partition of the nodes in two equally sized sets such that the number of edges between nodes in the different sets is minimized. More formally, the problem is to minimize

$$c(V_1, V_2) = |e(V_1, V_2)|, \text{ with } e(V_1, V_2) = \{(i, j) \in E : i \in V_1 \wedge j \in V_2\}, \quad (3.2)$$

where $c(V_1, V_2)$ is referred to as the cut size of the partition.

In the *quadratic assignment problem* (QAP), n facilities have to be assigned to n locations at minimum cost. Given a set $\Pi(n)$ of all permutations of $\{1, 2, \dots, n\}$ and two $n \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, the task is to minimize the quantity

$$C(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi(i)\pi(j)}, \quad \pi \in \Pi(n). \quad (3.3)$$

Matrix A can be interpreted as a distance matrix, i.e. a_{ij} denotes the distance between location i and location j , and B is referred to as the flow matrix, i.e. b_{kl} represents the flow of materials from facility k to facility l . The TSP and the GBP are special cases of the QAP.

The *NK-model* of Kauffman [16, 17] 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. The fitness f of a point $b = (b_1, \dots, b_N)$ is defined as follows:

$$f(b) = \frac{1}{N} \sum_{i=1}^N f_i(b_i, b_{i_1}, \dots, b_{i_K}), \quad (3.4)$$

where the fitness contribution f_i of gene i depends on the allele of gene b_i and the alleles of K other genes b_{i_1}, \dots, b_{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. The values for i_1, \dots, i_K are chosen randomly from $\{1, \dots, N\}$.

3.2.2 Fitness Landscape Definition

To define a fitness landscape for a given problem instance, a real valued fitness has to be assigned to each of the solutions $s \in S$ of the search space. Furthermore, we need to find

an arrangement of the solutions or genotypes in the genotypical space to form a landscape. The spatial structure of the landscape can be defined by a metric d , which assigns to each pair of solutions a distance value $d(\cdot, \cdot)$.

More formally, a fitness landscape of a problem instance for a given combinatorial optimization problem is a triple $\mathcal{L} = (S, f, d)$ and consists of a set of points (solutions) S , a fitness function $f : S \rightarrow \mathbb{R}$, which assigns a real-valued fitness to each of the points in S and a distance metric $d : S \times S \rightarrow \mathbb{R}$, for which it is required that

$$d(s, t) \geq 0, \quad d(s, t) = 0 \Leftrightarrow s = t, \quad d(s, t) \leq d(s, u) + d(u, t) \quad \forall s, t, u \in S.$$

Furthermore, $d_{min} \leq d(s, t) \leq d_{max} \quad \forall s, t \in S \wedge s \neq t$. The fitness landscape can be interpreted as a graph $G_{\mathcal{L}} = (V, E)$ with vertex set $V = S$ and edge set $E = \{(s, s') \in S \times S \mid d(s, s') = d_{min}\}$. The diameter of the landscape is the maximum distance between two points in the graph and is denoted $diam G_{\mathcal{L}}$, thus $d_{max} = diam G_{\mathcal{L}}$. The topology of the graph is, of course, problem dependent, e.g. for NK -landscapes the graph is a Hamming graph, for the graph bi-partitioning problem the graph is a Johnson graph, and for the traveling salesman problem the graph is a Cayley graph (see [35] for details).

For any instance of a given problem, there are many fitness landscapes, since many metrics can be defined on the set of all solutions to a given problem. The easiest and most straightforward definition of a distance function may be the following. Consider an elementary operator ω that transforms a solution s into a solution s' . The associated distance metric $d_{\omega}(s, t)$ is defined as the minimum number of applications of ω required to obtain t from s . Usually, the operator modifies a solution only slightly, e.g. by changing of a single gene of the genotype. For example, in binary coded problems, such an operator is the bit-flip operator which flips one bit at a time. To obtain one solution from the other, all differing bits have to be flipped, one by one. The number of times the bit-flip operator has to be applied is the number of different bits, and the distance metric induced by the operator is known as the Hamming distance $d_H(x, y) = \sum_i x_i \oplus y_i$ between bit vectors.

In the TSP, for example, an elementary operator is the edge-exchange operator which exchanges two edges contained in the current solution by two new edges maintaining feasibility. Applied twice, the operator generates a solution that has either zero, three or four edges not contained in the original tour, depending on the choice of the edges in the second application of the operator. If one or two edges are removed that have previously been inserted, the number of different edges is less than 4. So, instead of counting the number of applications of the edge-exchange operator, the number of different edges between two tours appears to be a better suited distance measure for the TSP.

3.2.3 Properties of Fitness Landscapes

For both performance prediction and memetic algorithm design, it becomes crucial to identify the characteristics of landscapes that have influence on the effectiveness of heuristic search methods.

The following properties of landscapes are known to have strong influence on heuristic search:

- the fitness differences between neighboring points in the landscape (ruggedness),
- the number of local optima (peaks in the landscape),
- the distribution of the local optima in the search space, and

- the topology of the basins of attraction of the local optima.

Several methods have been proposed to “measure” these properties. Some of them are presented in the following.

Autocorrelation Functions of Fitness Landscapes

The landscape ruggedness is an important property, since a smooth landscape where fitness varies only slightly along a random walk (the random application of an elementary operator) can be easily optimized by heuristic search algorithms. If small moves in a landscape lead to high fitness changes, the landscape is said to be rugged.

To measure the ruggedness of a fitness landscape, several methods have been proposed, for example the correlation functions proposed by Weinberger [38]. The *autocorrelation function* [34, 38] is defined as

$$\zeta(d) = \frac{\langle f(x)f(y) \rangle_{d(x,y)=d} - \langle f \rangle^2}{\langle f^2 \rangle - \langle f \rangle^2} \quad (3.5)$$

where $\langle x \rangle$ denotes the mean of all x_i ($\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$). It defines the correlation of points at distance d in the search space.

Alternatively, Weinberger suggested to use random walks to investigate the correlation structure of a landscape. The *random walk correlation function* [35, 36, 38]

$$r(s) = \frac{\langle f(x_t)f(x_{t+s}) \rangle - \langle f \rangle^2}{\langle f^2 \rangle - \langle f \rangle^2} \quad (3.6)$$

of a time series $\{f(x_t)\}$ defines the correlation of two points s steps away along a random walk through the fitness landscape. Based on these definitions, the correlation length ℓ [36] of the landscape can be defined as

$$\ell = -\frac{1}{\ln(|r(1)|)} = -\frac{1}{\ln(|\zeta(1)|)} \quad (3.7)$$

for $r(1), \zeta(1) \neq 0$. If the landscape is *statistically isotropic*, i.e. the time series $\{f(x_t)\}$ forms a stationary random process, then a single random walk is sufficient to obtain $r(s)$. If a time series is *isotropic, Gaussian and Markovian*, then the corresponding landscape is called AR(1) landscape, and the random walk correlation function is of the form $r(s) = r(1)^s = \exp(-s/\ell)$ with ℓ being the correlation length of the landscape. For example, AR(1) landscapes are found in the NK-model and the TSP [38]. Figure 3.1 shows $r(s)$ for an AR(1) landscape of the instance **tai100a** of the QAP.

The correlation length is a well suited measure for comparing landscapes for a given problem instance. The higher the correlation length, the smoother the landscape and hence the easier the search for an algorithm based on the underlying neighborhood of the landscape, since neighboring points have a higher correlation. Table 3.1 provides an overview of various fitness landscapes and their correlation lengths. These were obtained theoretically and have been confirmed by experiments [35, 36].

Fractal Landscapes

Another characteristic of fitness landscapes is the connection between correlation and self-similarity. A landscape is said to be fractal if the variance of the difference in fitness

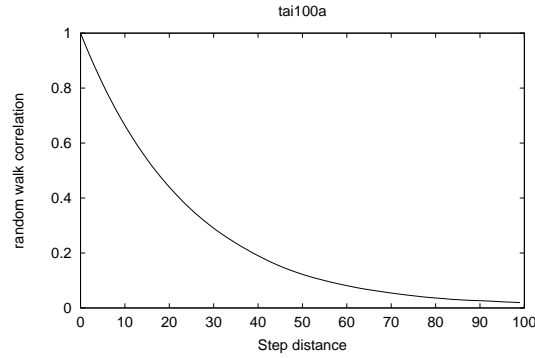


Figure 3.1: Random walk correlation function for the QAP instance `tai100a`

Table 3.1: Correlation in fitness landscapes of selected combinatorial optimization problems

Problem	Metric	diam G_L	$ \mathcal{N} $	ℓ
TSP	Edge-exchange	$n - 1$	$(n - 1)n/2$	$n/2$
	Node-exchange	$n - 1$	$(n - 1)n/2$	$n/4$
NK	Hamming	n	$n - 1$	$n/(k + 1)$
GBP	Exchanges	$n/2$	$(n - 1)n/2$	$\sim n/8$
	α -Flip	n	$n - 1$	$\sim n/4$

between two points in the landscape scales as a power law with their distance from each other. More formally, a landscape is fractal if $\langle |f(x) - f(y)|^2 \rangle \propto d(x, y)^{2h}$ for all pairs of points (x, y) in the search space. Examples of fractal landscapes with $h = 1/2$ include NK-Landscapes, TSP with edge-exchange, and the GBP.

Fractal landscapes can be separated into four classes according to Weinberger and Stadler [39]. However, it is still unknown how this classification is related to the performance of heuristic search methods.

Landscape Ruggedness and Epistasis

For many combinatorial optimization problems, under a suitable representation, the fitness or the cost function can be easily decomposed into fitness contribution functions for each gene of the genome representing a solution of the problem. Such a fitness decomposition allows us to identify the gene interactions of a given representation, or in other words, on which and how many other genes the fitness contribution of each gene depends. For example, if

$$f(x) = c \cdot \sum_{i=1}^n f_i(x_i, x_{i_1}, \dots, x_{i_{k(i)}}),$$

then the fitness contribution f_i depends on the allele (value) x_i of gene i and $k(i)$ other alleles at the loci $i_1, \dots, i_{k(i)}$. Thus, the values $k(i)$ determine the amount of interactions between the genes, called *epistasis*. In contrast to the model proposed by Davidor [6], who defines epistasis variance for functions where such a decomposition is not known or obvious

(for example, real valued function optimization using binary encodings), the amount of gene interaction is derived from the problem and the chosen representation itself, and there is no need for the use of statistical models.

Epistasis can be seen as a measure of problem difficulty since it determines the amount of nonlinearity of the problem. One would expect that for low epistasis search is easy: if there is no gene interaction, then there exists only one local optimum, but on the other hand, if the fitness contribution of a site depends on all other gene values, the fitness landscape becomes totally unstructured. It has been shown for NK -landscapes, where $K = k(i)$ determines the number of interacting genes, that there is one local optimum for $K = 0$, and the expected number of local optima for $K = N - 1$ is $\frac{2^N}{N+1}$. However, high epistasis (in terms of a high average number of interacting genes per locus) is not the only property making a problem hard to solve. For random geometric instances of the GBP, we have shown that search becomes easier with increasing epistasis [23], indicating that the number of the interactions is not the only important property. The gene interactions can be viewed as a directed graph (*graph of epistatic interactions*) with vertices representing the genes and edges defining the dependencies between the genes. An edge (i, j) indicates that the fitness contribution at gene j depends on the allele at gene i (f_j is of the form $f_j(\dots, x_i, \dots)$). Thus, the average vertex degree reflects the average number of interacting genes per locus. We have shown for the GBP that the structure of the graph of epistatic interactions has a high influence on the structure of the fitness landscape.

Fitness Distance Correlation

The *fitness distance correlation (FDC) coefficient* has been proposed in [15] as a measure for problem difficulty for genetic algorithms. The FDC coefficient ρ is defined as

$$\rho(f, d_{opt}) = \frac{\langle f d_{opt} \rangle - \langle f \rangle \langle d_{opt} \rangle}{(\langle f^2 \rangle - \langle f \rangle^2)(\langle d_{opt}^2 \rangle - \langle d_{opt} \rangle^2)} \quad (3.8)$$

and determines how closely fitness and distance to the nearest optimum in the search space are related. If fitness increases when the distance to the optimum becomes smaller, then the search is expected to be easy for selection-based algorithms: the optimum can successively be approached via fitter individuals. A value of $\rho = -1.0$ ($\rho = 1.0$) for a maximization (minimization) problem indicates that fitness and distance to the optimum are perfectly related and that search promises to be easy. A value of $\rho = 1.0$ ($\rho = -1.0$) means that with increasing fitness the distance to the optimum increases, too.

A *fitness distance plot* can be made to gain insight in the structure of the search space instead of simply calculating the correlation coefficient. The scatter plot contains much more information and can be interpreted easier than the coefficient and thus has been recommended for this type of analysis [15]. The scatter plot is generated by plotting the fitness of points in the search space against their distance to an optimum or best-known solution. Fitness distance analysis (FDA) has been applied by several researchers, including Kauffman [16] for NK -landscapes, Boese [4] for the TSP, Reeves for a flow-shop scheduling problem [32], Moscato for the binary perceptron problem [28], and Merz and Freisleben [23] for the GBP.

The disadvantage of the FDA is that the optimum solution has to be known in advance. In many cases, the best known solution can be used instead, since it is an optimum solution or it lies close to the optimum in the search space. For totally uncorrelated landscapes,

replacing the optimum by the best solution found may lead to a totally different fitness distance plot, and thus the FDA is of little help for performance prediction.

3.2.4 Local Search Neighborhoods

Fitness landscapes and local search algorithms are directly related to each other. Local search algorithms are characterized by the neighborhoods they depend on. For a given fitness landscape $\mathcal{L} = (S, f, d)$ the simplest neighborhood is defined by $\mathcal{N}(s) = \{s' \in S : d(s, s') = d_{min}\}$: $\mathcal{N}(s)$ is the set of all neighboring points of the solution s in the landscape \mathcal{L} . In other words, the neighborhood of a solution s is defined as the set of solutions that can be obtained by applying the elementary operator ω of the landscape once. More generally, the k -opt neighborhood is defined as

$$\mathcal{N}_{k-opt}(s) = \{s' \in S : d(s, s') \leq k\}. \quad (3.9)$$

The 3-opt neighborhood, for example, includes the 2-opt neighborhood and for $k = d_{max}$ the neighborhood is identical to the whole search space. The size of the k -opt neighborhoods $|\mathcal{N}_{k-opt}|$ grows exponentially in k for many combinatorial optimization problems, so usually only neighborhoods with small values for k are used in local search algorithms.

3.3 Memetic Algorithm Design

Evolutionary algorithms such as genetic algorithms [13], evolutionary programming [8], evolution strategies [31], and memetic algorithms [27] share the advantage that existing algorithms can be easily adapted to new problem domains. Only the problem specific details have to be rewritten, such as the evaluation of the fitness function, the definition of recombination and mutation operators as well as the population initialization function. All other parts of the algorithm do not have to be modified, for example the general framework, the parent selection mechanism, the replacement schemes and the data structures for storing the individuals of the population. The following steps are thus necessary to design a memetic algorithm for a new problem domain.

- Step 1:** Find a suitable representation for the candidate solutions of a given problem and an evaluation function for calculating the fitness of a given solution based on the representation.
- Step 2:** Find an efficient local search algorithm.
- Step 3:** Find a suitable initialization method for the initial population.
- Step 4:** Define the genetic mutation and recombination operators.

3.3.1 Representation and Local Search

Finding a suitable encoding for the solutions of the problem is tightly coupled with finding a good elementary operator for local search, since the elementary operator depends on the representation chosen. The elementary operator defines the landscape by inducing a distance metric over all pairs of points in the search space and thus is responsible for the performance of neighborhood based search. Generally, the elementary operator should

only change the elements or genes of a solution slightly, since this often results in a slight change of fitness. In some cases, however, it is necessary to change many of the genes to achieve a slight change in the fitness values. An example is the inversion operator on bit strings: here, the order of the bits between two positions is inverted.

To compare different combinations of operators and encodings and thus different landscapes for a given problem, an autocorrelation analysis can be carried out either mathematically or experimentally. The landscape with the higher correlation length (higher correlation of neighboring points in the search space) should be preferred over the other [27]. For example, consider two different fitness landscapes for the TSP as listed in Table 3.1. The landscape with the distance metric based on the elementary operator of exchanging two cities in the tour has a lower correlation length than the landscape based on the elementary operator of exchanging edges. It has been shown experimentally that a local search in the latter landscape is much more effective in finding near optimum solutions than a local search in the former landscape, see for example [33].

Another interesting example is the graph bi-partitioning problem. Enlarging the search space by allowing infeasible solutions leads to a smoother landscape if a suitable penalty function for reducing the fitness of infeasible solutions is incorporated. Angel and Zissimopoulos [1] have derived a penalty function for which the landscape (α -FLIP) has a higher correlation length and hence becomes smoother compared to the commonly chosen SWAP landscape (see Table 3.1). Another advantage of the FLIP landscape is that the neighborhood size $|\mathcal{N}|$ is lower and hence the time for searching the neighborhood for fitter individuals is reduced. Again, it has been shown experimentally that a local search based on the FLIP operator is more effective than a local search based on the SWAP operator [1].

Thus, autocorrelation analysis appears to be a suitable approach for comparing fitness landscapes considered in step 1 and 2.

Large Neighborhoods

Although the smallest possible neighborhood for a combinatorial problem is preferable, there are cases where larger neighborhoods can be searched very efficiently. Consider the k -opt neighborhoods \mathcal{N}_{k-opt} . All solutions in these sets cannot be examined during the search, but there are algorithms that are searching a small subset and are highly effective. For example, the Lin-Kernighan heuristic [20] for the TSP and the Kernighan-Lin heuristic [18] for the GBP exchange a variable number of edges and a variable size of subsets, respectively. Since in case of the TSP an exchange of k edges is realized by performing a sequence of exchanges of two edges, the same distance measure can be used as for 2 -opt. Analogously, for the GBP the same distance measure is recommended as for the SWAP operator. To speed up the running times for k -opt search algorithms, special data structures are necessary for large problem sizes. For the TSP, several data structures have been investigated [9], and for the GBP, a data structure developed by Fiduccia and Mattheyses [7] increases the efficiency of the Kernighan-Lin heuristic tremendously.

3.3.2 Generating Starting Solutions

In evolutionary algorithms, the starting solutions are usually generated in a purely random fashion. Hence, it is straightforward to do this in memetic algorithms, too. To operate on locally optimal solutions, after generation of the starting solutions a local search is applied to each solution. However, to exploit the structure of the problem, randomized

heuristics can be used that (a) produce a large variety of solutions, and (b) are good for a combination with local search. The advantage is two-fold: first, the combination of a construction heuristic and local search may produce better solutions, and second, the combination may be faster than applying local search to randomly selected points in the landscape, since the number of iterations of the local search is reduced due to the much better starting point generated by the heuristic.

Greedy heuristics are often good candidates for the construction of feasible solutions. Viewing a solution of a combinatorial optimization problem as a genome of length $|x| = n$ (n denoting the problem size), a greedy heuristic determines in each step an allele value for a selected location in a genome without violating the feasibility constraints of the problem. After n steps, all genes have their values assigned and a feasible solution is constructed. Which allele and which location is chosen depends on a greedy selection criterion aimed to optimize the objective function.

Epistasis and Greedy Heuristics

The effectiveness of a greedy heuristic depends highly on the non-linearity of the problem. If epistasis is high, then the choice of a gene value at a step in the construction heuristic strongly influences the future choices of the remaining locations in the genome that have not been assigned. A wrong greedy choice at an early stage of the algorithm has fatal consequences and may lead to a low fitness. On the other hand, if epistasis is low, an unfavorable decision has a much smaller influence on the overall fitness of the solution being built, since most decisions in the future of the algorithm are not affected. In the extreme case where no interactions are present and hence the choice of a gene value in each step is independent of the choices in the other steps, the greedy algorithm is able to find the global optimum. Thus, for problems with low epistatic interactions, a greedy algorithm is preferable to a purely random generation of solutions.

The TSP is a good example where a greedy heuristic works extremely well in combination with local search. Due to the feasibility constraints of the problem, in the last steps of the algorithm arbitrarily long edges are included in the tour constructed, but a local search is able to compensate this effect. The greedy heuristic [19] has been shown to be superior to all other heuristics when combined with local search [14, 33]. Furthermore, the number of iterations of the local search is reduced compared to random starting tours, since many short edges are already contained in the tour and have not to be discovered by local search.

Fitness Distance Analysis

The initial population is important for the performance of a population based algorithm, so special attention has to be paid when creating it. To investigate how the points represented by the initial population are distributed in the search space relative to the optimum solution, an analysis of the fitness distance correlation can be performed. The fitness distance plots can help in comparing heuristic construction algorithms and local search, and can provide hints of how the memetic algorithm will perform. For example, the fitness distance plots of some GBP instances show the effectiveness of a greedy heuristic called *Differential Greedy* [2] compared to the Kernighan-Lin local search on structured graphs (instances with low epistasis), as shown in Figure 3.2.

Instead of the fitness, the cut size difference $\Delta c(x) = c(x) - c_{opt}$ is plotted in the figure (note that the GBP is a minimization problem). The plots on the left representing 2500

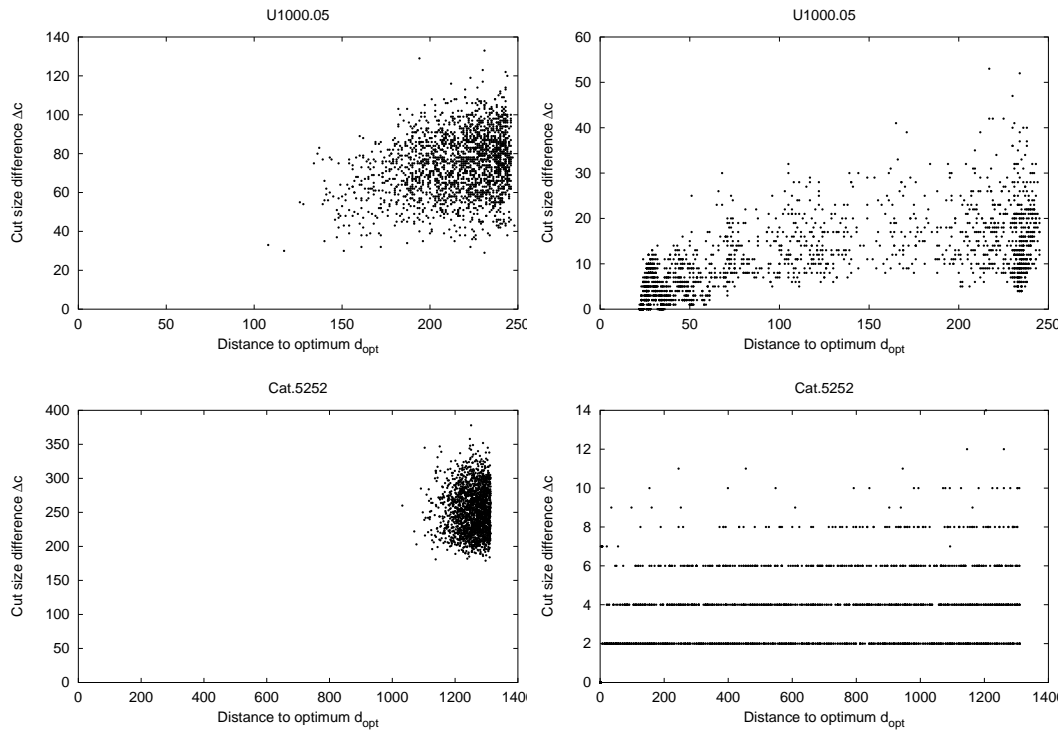


Figure 3.2: Cut size difference vs. distance to optimum for Kernighan-Lin local optima (left) and Diff-Greedy solutions (right)

local optima produced by the Kernighan-Lin heuristic show a much lower correlation than the plots on the right produced by the differential greedy heuristic. In case of graph *Cat.5252* there is *no* correlation of the local optima. In both cases, the objective values of the solutions produced by differential greedy are much closer to the optimum than in case of the Kernighan-Lin solutions.

But there are also graphs for which the greedy heuristic does not perform better than the local search, as shown in Figure 3.3. The first graph *G1000.0025* is an unstructured random graph. Here, Kernighan-Lin solutions and differential greedy solutions are distributed similarly in the search space. The second graph is a random geometric graph *U1000.40* with high epistasis. Here, only few Kernighan-Lin local minima exist and the chance to hit the global optimum is high, thus the Kernighan-Lin heuristic appears to be superior to the greedy approach.

Since the greedy strategy is good for problems with low epistasis and the Kernighan-Lin approach is better for structured problems with high epistasis, the combination of both is a good way to produce the initial population for the GBP.

As shown above, fitness distance analysis can help in finding the best initialization methods for a given problem. However, to compare algorithms, the analysis must be performed for each problem instance, and the optimum solution or a near optimum solution must be known.

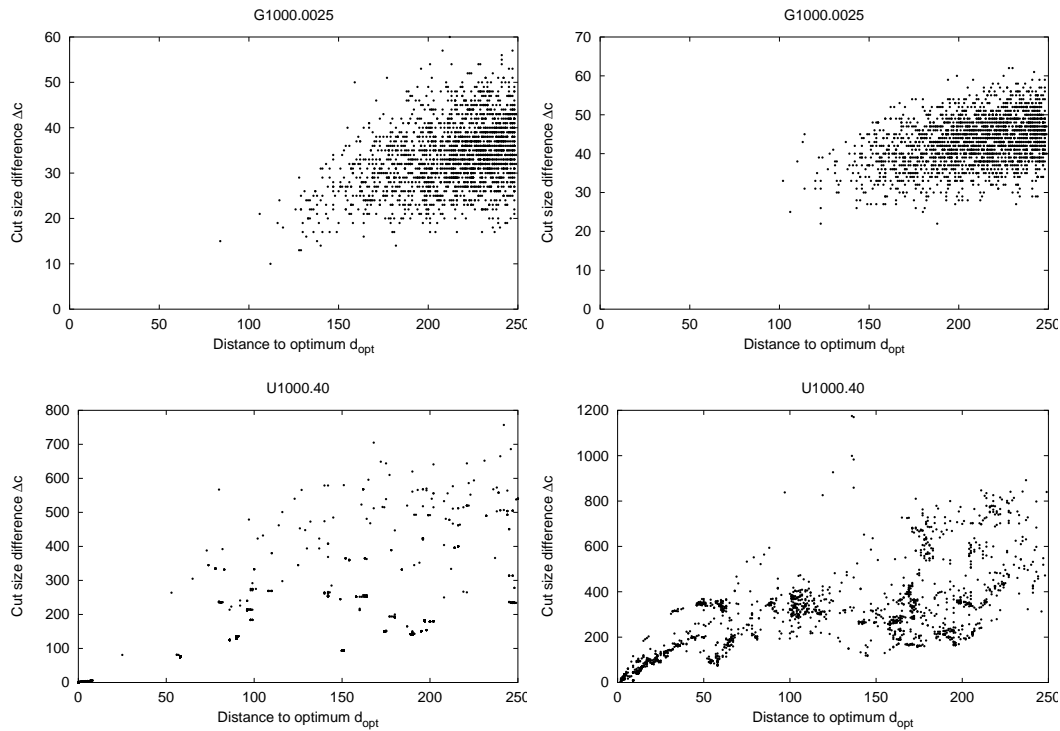


Figure 3.3: Cut size difference vs. distance to optimum for Kernighan-Lin local optima (left) and Diff-Greedy solutions (right)

3.3.3 Genetic Operators

In a memetic algorithm framework, mutation and recombination operators act as diversification strategies. By utilizing the information contained in the population – the individuals in the population may be located in a region of the search space containing local optima with high fitness – new starting points for a local search have to be discovered leading to even better local optima. Recombination and mutation operators hence perform jumps in the search space, leading from a local optimum away to a new point from which the next hill is climbed or (for minimization) the next valley is explored, as illustrated in Figure 3.4 for the TSP.

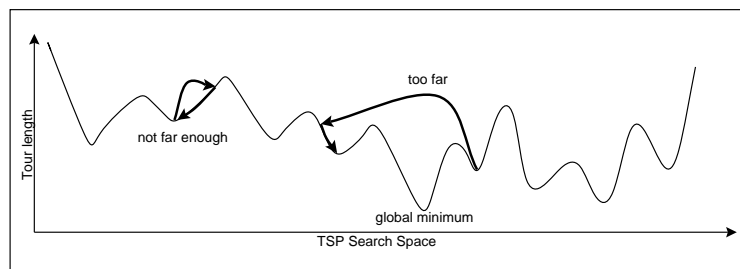


Figure 3.4: Diversification by jumping in the search space

This implies that the distribution of the local optima has to be somehow structured. The

design of mutation and recombination operators is discussed in the following.

Mutation

Mutation is an unary operator and has to be performed so that the subsequently applied local search does not revisit the parent solution by falling back into the same local optimum. The optimal jump distance depends at least on one property of the search space: the size of the attractor region of the current local optimum. For some landscapes it has been shown that the correlation length ℓ of a landscape is related to the average distance between local optima: in these landscapes there is one local optimum in a ball of radius $R(\ell)$, where $R(s)$ denotes the expected distance of the start and end point of a simple random walk of s steps. Since the direction of the jump is random, no other properties can be exploited with the unary operator.

Recombination

Recombination is a binary operator, and hence a jump can be performed with a predefined direction. In genetic algorithms, recombination is performed in analogy to biology by crossing over two parent bit vectors at a randomly selected position. Another widely used recombination mechanism is known under the name uniform crossover. Uniform crossover is a more general form of recombination, and single-point or k -point crossover are special cases [37]. These crossover techniques applied to binary vectors have the following properties:

- (i) The bit values for the locations that are identical in both parents are preserved in the offspring.
- (ii) The hamming distance d_H between the parents x and y and the offspring z are lower or equal to the distance between the parents. Furthermore, $d_H(x, z) + d_H(z, y) = d_H(x, y)$.

For permutation search spaces, the first property can easily be fulfilled while the second can not. However, if (i) is obeyed, the following holds for all parents x and y and offspring z : $d(z, x) \leq d(x, y)$ and $d(z, y) \leq d(x, y)$, and furthermore $d(x, z) + d(z, y) \leq 2 \cdot d(x, y)$. Following Radcliffe's and Surry's terminology [30, 29], a recombination operator obeying (i) is called *respectful*, while an operator obeying (ii) is called *assorting*. Recombination operators which fulfill (i) or even (ii) produce offspring that are contained in a region of the search space spanned by the two parents. The size of the region decreases during evolution since the individuals of the population move closer together from generation to generation. Hence, the direction of the jumps produced by respectful recombination operators is oriented towards a region between the parents, and the jump distance changes dynamically during the search.

Fitness Distance Analysis

To address the question of how a distance fitness analysis can help in designing genetic operators, it is instructive to look at some very different fitness landscapes. In Figure 3.5 plots are provided for landscapes of several combinatorial problems.

The first two plots (Cat.5252 and Breg5000.16) are taken from the analysis of instances of the GBP, and the second two are results of a fitness distance analysis for a TSP (att532)

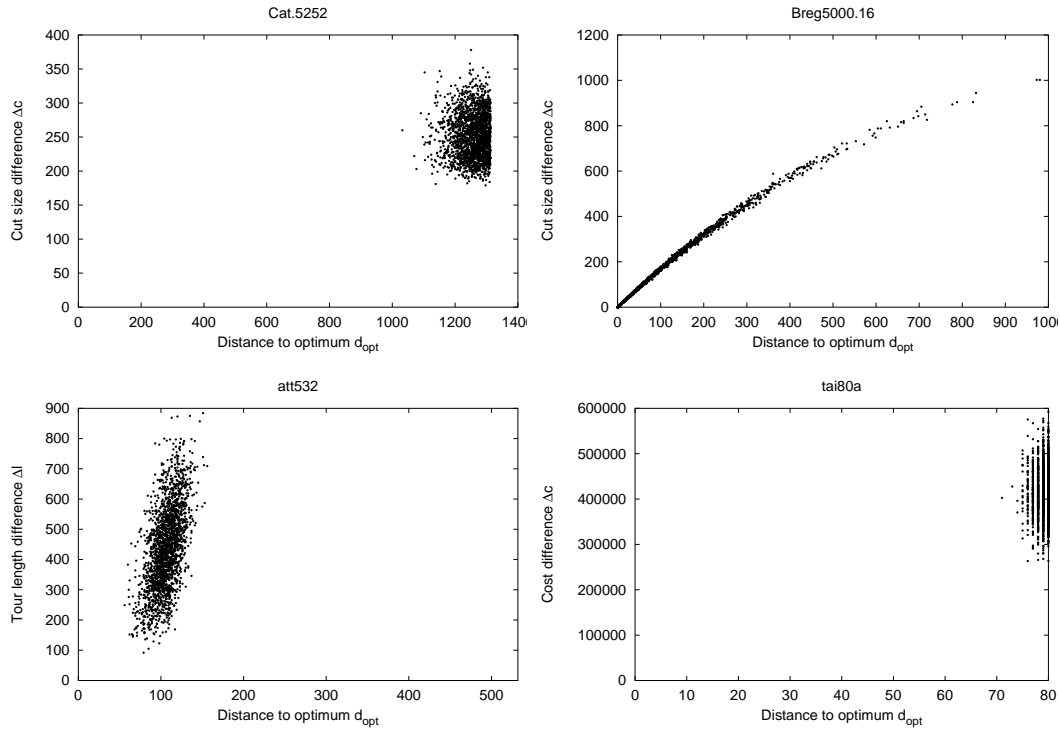


Figure 3.5: Correlation of local optima for various combinatorial optimization problems

and a quadratic assignment problem (QAP) (**tai80a**) instance, respectively. In all plots, 2500 local optima are displayed relative to the optimum or best-known solutions for the problem. While the first plot (**Cat.5252**) does not show correlation between fitness and distance, the second one (**Breg5000.16**) shows perfect correlation ($\rho = 0.02$ and $\rho = 0.99$, respectively). In the first case, the local optima have an average distance of nearly the maximum distance of the search space between each other. Hence, the local optima are uniformly distributed in the search space. Regarding the distribution of local optima, the search space does not exhibit a structure that could be exploited. The second landscape of the GBP is ideal for memetic algorithms. There is a “path” to the global optimum from all local optima via fitter local optima. With increasing fitness (decreasing cut size), the local optima are closer to the global optimum. The fitness distance plot for the TSP landscape (**att532**) shows a correlation between fitness and distance to the optimum. Furthermore, the local optima are found in a small fraction of the search space, and they appear to be relatively close together. In this case, the maximum distance between two local optima is smaller than $1/3$ of the maximum distance of the landscape. This landscape and the second landscape of the GBP (**Breg5000.16**) exhibit a structure that is called the *big valley* structure [4]: the global optimum lies more or less central in the subspace containing the local optima. The last plot shows the relation between fitness and distance to the optimum of a typical QAP instance (**tai80a**). Here, points with optimum or near optimum fitness are arbitrarily far away from each other and most of the solutions have maximum distance to the optimum.

Choice of Operators

If the landscape is structured, recombination operators are preferable over mutation operators since they are able to exploit the structure. Big valley characteristics are best exploited by directed jumps performed by respectful recombination operators, while these operators are not effective if the structure is missing. We have shown that for NK -landscapes with high epistasis mutation becomes favorable over crossover [24]. As shown in [26], for unstructured GBP instances (high epistasis) this also holds. We have made additional experiments for the TSP and for the QAP to show the relation between landscape structure and operator effectiveness. Table 3.2 shows the results for the TSP and Table 3.3 for the QAP, respectively. For the TSP, the average number of generations (gen) and the average time in seconds (t/s) to reach the optimum in 30 of 30 runs is provided. For the QAP, the average number of generations (gen), the best cost found averaged over 30 runs, and the time in seconds after the algorithm was terminated is given.

Table 3.2: Crossover vs. mutation based MA on 4 TSP instances

Instance	tour length	DPX		Mutation	
		gen	t/s	gen	t/s
lin318	42029 (opt)	68	25	31	38
pcb442	50778 (opt)	99	35	92	63
att532	27686 (opt)	452	131	280	230
rat783	8806 (opt)	156	61	217	88

As expected, for the TSP, recombination is preferable to mutation, while for the QAP mutation outperforms recombination-based search.

Table 3.3: Crossover vs. mutation based MA on 5 QAP instances

Instance	DPX		Mutation		t/s
	gen	avg. cost (quality)	gen	avg. cost (quality)	
sko100a	16667	152804.6 (0.53%)	993	152210.4 (0.14%)	1800
tai100a	765	21667111.8 (2.56%)	1551	21366635.6 (1.14%)	1800
tai150b	244	504998950.4 (1.22%)	629	500976809.6 (0.42%)	3600
tho150	8936	8161583.4 (0.34%)	718	8150674.6 (0.21%)	3600
tai256c	875	44858227.6 (0.22%)	3431	44785102.8 (0.06%)	3600

In both cases, the recombination operator is the distance preserving crossover operator DPX [11, 21]. The operator is respectful but also highly disruptive: the distance of the jumps performed in the search space equals the distance between the two parents. The local search algorithm used for the TSP is the Lin-Kernighan heuristic [20] and a simple 2-*opt* local search has been used for the QAP. Mutation is performed with a non-sequential four-change [20] in case of the TSP, and with a random mutation of jump distance $d = 30$ in case of the QAP.

3.4 Memetic Algorithm Performance

Memetic algorithms, in particular evolutionary algorithms incorporating local search, have been shown to outperform many other heuristic search algorithms for various problems: for the TSP, our memetic algorithm, also called genetic local search, has been shown to be one of the most effective algorithms [3, 10, 22] – a predecessor of our improved approach has won the first international contest on evolutionary optimization (1st ICEO) [3]. The results presented in Table 3.2 are even better than previously published [22]. In case of the QAP, our approach works extremely well and appears to be superior to tabu search, ant colonies, simulated annealing and also scatter search [21, 25, 5]. For NK -landscapes we have shown that genetic local search is superior to genetic algorithms and multi-start local search [24]. Recently, we have shown for the GBP that our memetic algorithm is superior to other hybrid evolutionary approaches, simulated annealing, and in almost all cases superior to tabu search [26].

3.5 Conclusions

In this chapter, we have presented techniques to analyze combinatorial optimization problems to design highly effective search algorithms with special emphasis on memetic algorithms. The presented autocorrelation functions of fitness landscapes are well suited to determine the local structure of a landscape and thus help in finding preferable representations and local search neighborhoods for a memetic algorithm. In particular, the correlation length of a landscape has been shown to be a good indicator for the effectiveness of representations in combination with local search. The correlation length of a landscape is based on the correlation of neighboring points in the search space and can be determined either mathematically or experimentally. The higher the correlation length, the better the performance of a local search.

On the other hand, fitness distance analysis (FDA) is suited for analyzing the global structure of a landscape. The distribution of local optima is important in memetic algorithm design because if a certain structure is present, it should be exploited by the genetic operators used in the approach. The conducted experiments have shown that if fitness and distance to the optimum are correlated, recombination operators are preferable to mutation operators. If no structure in the distribution can be observed, a mutation-based approach appears to be superior to recombination.

Furthermore, FDA can be utilized to find suitable heuristics for initialization of the population. FDA allows a comparison of heuristics in terms of ability to adequately sample the search space and to predetermine promising features of solutions to a problem.

However, our studies of landscapes of the graph bi-partitioning problem have shown that it is dangerous to perform FDA for only one type of instance of a given optimization problem. The results may not be generalized to other types of instances: some instances of the graph bi-partitioning problem show a high correlation between fitness and distance to the optimum, while for others no correlation could be observed.

Epistasis has been considered as an indicator of the hardness of an optimization problem. Our studies on the graph bi-partitioning problem have shown that the concept of modeling gene interactions by a *graph of epistatic interactions* can help in understanding the influence of epistasis on heuristic search.

Future work should address the question of how to design recombination operators that can best exploit certain properties of landscapes. Furthermore, it is desirable to be able to

predict the optimum jump distances for mutation operators in unstructured landscapes. Here, we believe the correlation length of a landscape can be utilized beneficially.

Other statistical properties may be useful to predict the performance of memetic algorithms. For example, it should be investigated how the classification of fractal landscapes proposed by Weinberger and Stadler [39] can help in classifying problems in terms of hardness for optimization algorithms.

Bibliography

- [1] E. Angel and V. Zissimopoulos, “Autocorrelation Coefficient for the Graph Bipartitioning Problem,” *Theoretical Computer Science*, vol. 191, pp. 229–243, 1998.
- [2] R. Battiti and A. Bertossi, “Differential Greedy for the 0–1 Equicut Problem,” in *Proceedings of the DIMACS Workshop on Network Design: Connectivity and Facilities Location*, (D. Du and P. Pardalos, eds.), American Mathematical Society, 1998. to appear.
- [3] H. Bersini, M. Dorigo, S. Langerman, G. Seront, and L. Gambardella, “Results of the First International Contest on Evolutionary Optimisation (1st ICEO),” in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, (Nagoya, Japan), pp. 611–615, 1996.
- [4] K. Boese, “Cost versus Distance in the Traveling Salesman Problem,” Tech. Rep. TR-950018, UCLA CS Department, 1995.
- [5] V.-D. Cung, T. Mautor, P. Michelon, and A. Tavares, “A Scatter Search Based Approach for the Quadratic Assignment Problem,” in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC)*, (T. Baeck, Z. Michalewicz, and X. Yao, eds.), (Indianapolis, USA), pp. 165–170, IEEE Press, 1997.
- [6] Y. Davidor, “Epistasis Variance: Suitability of a Representation to Genetic Algorithms,” *Complex Systems*, vol. 4, no. 4, pp. 369–383, 1990.
- [7] C. M. Fiduccia and R. M. Mattheyses, “A Linear-Time Heuristic for Improving Network Partitions,” in *Proceedings of the 19th ACM/IEEE Design Automation Conference DAC 82*, pp. 175–181, 1982.
- [8] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York: John Wiley & Sons, 1966.
- [9] M. L. Fredman, D. S. Johnson, L. A. McGeoch, and G. Ostheimer, “Data Structures for Traveling Salesmen,” *Journal of Algorithms*, vol. 18, pp. 432–479, 1995.
- [10] B. Freisleben and P. Merz, “A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems,” in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, (T. Bäck, H. Kitano, and Z. Michalewicz, eds.), pp. 616–621, IEEE Press, 1996.
- [11] B. Freisleben and P. Merz, “New Genetic Local Search Operators for the Traveling Salesman Problem,” in *Proceedings of the 4th International Conference on Parallel*

- Problem Solving from Nature - PPSN IV*, (H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, eds.), pp. 890–900, Springer, 1996.
- [12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
- [13] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [14] D. S. Johnson and L. A. McGeoch, “The Traveling Salesman Problem: A Case Study,” in *Local Search in Combinatorial Optimization*, (E. H. L. Aarts and J. K. Lenstra, eds.), pp. 215–310, Wiley and Sons, New York, 1997.
- [15] T. Jones and S. Forrest, “Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms,” in *Proceedings of the 6th International Conference on Genetic Algorithms*, (L. J. Eshelman, ed.), pp. 184–192, Morgan Kaufmann, 1995.
- [16] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- [17] S. A. Kauffman and S. Levin, “Towards a General Theory of Adaptive Walks on Rugged Landscapes,” *Journal of Theoretical Biology*, vol. 128, pp. 11–45, 1987.
- [18] B. Kernighan and S. Lin, “An Efficient Heuristic Procedure for Partitioning Graphs,” *Bell Systems Journal*, vol. 49, pp. 291–307, 1972.
- [19] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. New York: Wiley and Sons, 1985.
- [20] S. Lin and B. Kernighan, “An Effective Heuristic Algorithm for the Traveling Salesman Problem,” *Operations Research*, vol. 21, pp. 498–516, 1973.
- [21] P. Merz and B. Freisleben, “A Genetic Local Search Approach to the Quadratic Assignment Problem,” in *Proceedings of the 7th International Conference on Genetic Algorithms*, (T. Bäck, ed.), pp. 465–472, Morgan Kaufmann, 1997.
- [22] P. Merz and B. Freisleben, “Genetic Local Search for the TSP: New Results,” in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, (T. Bäck, Z. Michalewicz, and X. Yao, eds.), pp. 159–164, IEEE Press, 1997.
- [23] P. Merz and B. Freisleben, “Memetic Algorithms and the Fitness Landscape of the Graph Bi-Partitioning Problem,” in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN V*, (A.-E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, eds.), pp. 765–774, Springer, 1998.
- [24] P. Merz and B. Freisleben, “On the Effectiveness of Evolutionary Search in High-Dimensional *NK*-Landscapes,” in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, (D. Fogel, ed.), pp. 741–745, IEEE Press, 1998.
- [25] P. Merz and B. Freisleben, “Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem,” Tech. Rep. 99-02, University of Siegen, Germany, 1999.

- [26] P. Merz and B. Freisleben, "Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning," *Evolutionary Computation*, 1999, to appear.
- [27] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," Tech. Rep. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
- [28] P. Moscato, "An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: The Role of Tabu Search," *Annals of Operations Research*, vol. 41, no. 1-4, pp. 85–121, 1993.
- [29] N. Radcliffe and P. Surry, "Fitness Variance of Formae and Performance Prediction," in *Proceedings of the Third Workshop on Foundations of Genetic Algorithms*, (L. Whitley and M. Vose, eds.), (San Francisco), pp. 51–72, Morgan Kaufmann, 1994.
- [30] N. Radcliffe and P. Surry, "Formal Memetic Algorithms," in *Evolutionary Computing: AISB Workshop*, (T. Fogarty, ed.), pp. 1–16, Springer-Verlag, Berlin, 1994.
- [31] I. Rechenberg, *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.
- [32] C. R. Reeves, "Landscapes, Operators and Heuristic Search," *Annals of Operations Research*, 1998. To appear.
- [33] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*. Vol. 840 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, 1994.
- [34] P. F. Stadler, "Correlation in Landscapes of Combinatorial Optimization Problems," *Europhys. Lett.*, vol. 20, pp. 479–482, 1992.
- [35] P. F. Stadler, "Towards a Theory of Landscapes," in *Complex Systems and Binary Networks*, (R. Lopéz-Peña, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche, eds.), (Berlin, New York), pp. 77–163, Springer Verlag, 1995. SFI preprint 95-03-030.
- [36] P. F. Stadler, "Landscapes and their Correlation Functions," *J. Math. Chem.*, vol. 20, pp. 1–45, 1996. SFI preprint 95-07-067.
- [37] G. Syswerda, "Uniform Crossover in Genetic Algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 2–9, Morgan Kaufman, 1989.
- [38] E. D. Weinberger, "Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference," *Biological Cybernetics*, vol. 63, pp. 325–336, 1990.
- [39] E. D. Weinberger and P. F. Stadler, "Why Some Fitness Landscapes are Fractal," *J. Theor. Biol.*, vol. 163, pp. 255–275, 1993.
- [40] S. Wright, "The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution," in *Proceedings of the Sixth Congress on Genetics*, p. 365, 1932.