18 FITNESS LANDSCAPES AND PERFORMANCE OF META-HEURISTICS

Cyril Fonlupt[1], Denis Robilliard[1],
Philippe Preux[1] and El-Ghazali Talbi[2]

[1]LIL, Université du Littoral-Côte d'Opale
BP 719, 62228 Calais Cedex, France.
fonlupt@lil.univ-littoral.fr

[2]LIFL, Université de Lille 1, Bâtiment M3
59655 Villeneuve d'Ascq Cedex, France.
talbi@lifl.fr

Abstract: We perform a statistical analysis of the structure of the search space of some planar, euclidian instances of the traveling salesman problem. We want to depict this structure from the point of view of iterated local search algorithms. The objective is two-fold: understanding the experimentally known good performance of metaheuristics on the TSP and other combinatorial optimization problems; designing new techniques to search the space more efficiently. This work actually led us to design a hybrid genetic algorithm that competes rather well with other local search heuristics for the TSP, notably Jünger's et al.'s version of ILK. This work also opens promisful horizons to the study of other combinatorial optimization problems such as the quadratic assignment problem.

18.1 INTRODUCTION

Metaheuristics have shown their ability to reach good approximations of optimal solutions to numerous $\mathcal{NP}$-hard problems. Some authors have tried to explain these performances by a study of the topology of the research space associated to the problem (see e.g. [13, 6, 31]). We wish to go on further on this kind of studies and use them to design better search methods.

Metaheuristics come in different brands, such as simulated annealing [18], tabu search [8, 9], evolutionary algorithms (including all variants such as genetic algorithms [10], evolution programming [26]), ant colony systems [5] being the most famous. (See [29, 1] for recent reviews of metaheuristics.) Basically, metaheuristics are composed of a loop. The body of the loop is based on the following elements:

- a current state which is a point, or a collection of points of the research space,
- a neighborhood defined by some operator Ω that is applied on the current state to get the set of potential next states. To cope with the case of algorithms involving more than one operator such as evolutionary algorithms, we consider in this case that Ω is the composition of the different operators. This operator can be quite complicated. It might even be a metaheuristic.
- a transition rule (also called a pivoting rule, or a selection strategy) which chooses the actual next state among the set of potential next states,
- a halting criterion to decide when to stop the algorithm.

The initial state is either taken at random in the research space, or obtained from a construction heuristic.

The theoretical background of local search is not yet sheding much light on their actual behavior [35, 36]. Given a starting point, we can neither predict the number of iterations that will be done before reaching a local optimum, nor predict the quality of this local optimum. Assuming $\mathcal{P} \neq \mathcal{NP}$, the number of iterations might grow exponentially with the size of the problem, and there are local optima of arbitrary bad quality. We have no formal result about the proportion of local optima, nor the number of global optima, nor their distribution in the search space.

However, experimental studies have shown that the situation is not so hopeless in real applications. Indeed, in the case of the Traveling Salesman Problem (TSP), metaheuristics are providing excellent local optima: for instances of up to 10^5 cities, solutions which length are not more than 1% in excess to that of the shortest tour length are reached within minutes on typical workstations [14]. Metaheuristics do also perform very well on problems such as vehicle routing problem, quadratic assignment problem, graph coloring problem, set partitioning problem, VLSI layout design, ...

Facing these experimental evidences, we have felt that it would be useful to understand the performance of metaheuristics. To meet this goal, we study the dynamics of a great number of walks of a very simple metaheuristic as it is solving different instances of a problem. This metaheuristic is an ascent walk (or descent whether maximizing or minimizing the objective function): in the previous algorithmic schema, the transition rule either deterministically chooses the best next state (SDW: Steepest Descent Walk), or one better neighbor (RDW: Random Descent Walk). In both cases, the walk stops when no more improvement is possible from the current state. The current state is then a local optimum for Ω. Though very simple, this algorithm is at the basis of metaheuristics, so that our study is relevant with regards to the goal. Both tabu search and simulated annealing have much in common with these algorithms, though the pivoting rule is not so crude in their cases. The metaheuristic is probing the search space and we collect information along the walk in order to figure out what the landscape looks like. In this paper, we study the TSP. We observe that all walks reach a certain, very small, area of the search space, where lots of local optima are concentrated around one, or more, shortest tour(s). In itself, this result is not new since the discovery of a "deep valley" by S. Kirkpatrick and G. Toulouse in 1985 [19]. However, the way we obtain this result is very different, relying on statistics. One may feel that this approach is very weak and does not prove anything. Though perfectly aware of this weakness, we think that the fact that we obtain the same conclusions as earlier works is re-assuring. We also get some confidence from the fact that when we use this knowledge to derive a new search algorithm to take advantage of it, we obtain an algorithm which performance can be compared with some of the best heuristics to date, such as ILK. Furthermore, we have since then used the same approach to study the structure of an other combinatorial problem (the Quadratic Assignment Problem). Again, we are able to use the knowledge of the structure of the problem to improve search algorithms. So we feel confident that this approach to study the structure of search space can be used to tackle many combinatorial optimization problems, even if one has to be very careful with the conclusions he/she can draw from it.

E. Weinberger [32] has initiated studies on random walks (in a random walk, the pivoting rule takes a neighbor at random) in various search spaces. Though this algorithm is completely useless for the purpose of optimization, the analysis of these walks is able to bring some interesting information about the structure of the space, the distribution of local optima and their quality. We consider this first work, its followers (such as [33, 30, 12]), and some other works dealing with the study of the correlation of the cost of neighborhing points [25, 11] as complementary to ours.

In the following, we first present our study of the structure of the planar, euclidian TSP. From it, we derive a search algorithm and provide experimental results and comparisons with other metaheuristics in Section 18.3. We conclude by a discussion and perspectives of this work.

18.2 THE STRUCTURE OF THE PLANAR EUCLIDIAN TSP

18.2.1 The Traveling Salesman Problem

The Traveling Salesman Problem has a very long history. Though, as it is now suspected, the TSP is rather atypical in its structure, it is often used due its intuitive appeal. We define it very briefly, pointing the reader to Lawler et al.'s TSP book [20] for (much) more precisions, and also the book from Jünger et al.'s [16].

Given a set of n cities $c_{i,0\leq i<n}$ in a d dimensional space, one has to find the shortest tour that goes through each city. From any city, all other cities may

be reached. The problem may be defined either by the coordinates of the cities, or by the distance matrix that provides the distance between any two cities. n is the size of the instance.

Obviously, the shortest tour is one that visits each town once and only once. Thus, it is a hamiltonian tour.

In this paper, we restrict ourselves to the study of planar, symetric TSP where the distance between cities is euclidian. Most studies of the TSP are limited in this manner as most applications of the TSP can be seen as two, or three, dimensional in our physical world. Even in this case, TSP is $\mathcal{NP}$-hard [7]. We name 2ETSP this class of instances of the TSP.

A tour is coded by the set of n edges that it is composed of. There are $N = (n-1)!/2$ different tours when taking account of symetries and redundancies in the coding of solutions. This is also the number of points in the search space for the algorithms that are considered in this paper.

The basic move of local search is the 2-opt [4] move which consists in removing two edges from a tour and reconstruct a new different tour (there is always one and only one way to reconstruct a new valid tour that is different from the original). The 3-opt proposed by Bock in 1958 exchanges up to 3 edges [21]. We can go on and define the k-opt which exchanges k edges. In practice, values of $k > 4$ are seldomly used. The LK-move, defined by Lin and Kernighan in 1973 [22], is based on the exchange of edges, the number of edges to exchange being evaluated to get the best possible improvement. Since 1973, it is the best move known for the TSP and is at the core of the most successful algorithms to date for the TSP. We will not go further in the description of the LK move and refer the interested reader to [14, 15].

We define two distances to measure the difference between two tours. First, we define an intuitive distance:

Definition 1 *Let $t1$ and $t2$ two tours. We denote $\delta(t1, t2)$ the number of edges shared by both $t1$ and $t2$.*

[19] already used such a distance, though normalized by dividing it by n, the size of the instance.

Obviously, we have: $0 \leq \delta(t1, t2) \leq n$ for any two valid tours for an instance of size n.

We would like the distance between two tours not to be merely intuitive, but related to the operator used in the search algorithm. This distance encompasses some sort of "minimal algorithmic difficulty" to obtain a point out of a given point. The definition is stated having 2-opt in mind. It can be straightforwardly modified to deal with any other operator.

Definition 2 *Let $t1$ and $t2$ two tours. We denote $\Delta_{2opt}(t1, t2)$ the minimum number of applications of 2-opt to obtain $t2$ out of $t1$.*

Obviously, Δ_{2opt} is symetric, that is $\Delta_{2opt}(t1, t2) = \Delta_{2opt}(t2, t1)$.

This distance is very attractive but we know of no polynomial way to compute it. So, we use δ as an approximation of Δ as did [19, 2]. [2] prooves the following inequalities: $\delta(t1, t2)/2 \leq \Delta_{2opt}(t1, t2) \leq \delta(t1, t2)$.

This all implies that when in the sequel we say that two tours are close to each others (that is, their distance δ is much smaller than n), then we can go from one to the other in a very small amount of iterations (again, small with regards to the size of the instance).

18.2.2 The landscape of the 2ETSP

The landscape is involved by the operator Ω, that is, the definition of neighborhood used in the algorithm. The walk itself is also due to the pivoting rule.

Let us first define the notion of landscape (also called fitness landscape, energy landscape, or cost surface). Let $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ be the graph which vertices $\mathcal{V}$ are the N points of the search space. The edges of the set $\mathcal{E}$ connect all couple of points (x, y) such that $x \in \Omega(y)$, or $y \in \Omega(x)$, that is two points are connected if one can be obtained by the application of the operator on the other. Then, considering this graph as the ground floor, we elevate each point x to an altitude $z(x)$ equal to its cost, the length of x in our case. We obtain a surface, or landscape, made of peaks, valleys, plateaus, cliffs, ... The search algorithm aims at finding the lowest point(s). Local optima are points such that all their neighbors are higher than them. The problem lies in the difficulty to have a realistic view of this landscape: it is a geometrical object of N dimensions, which turns to be many thousands even for very moderate values of n. Thus, one has to devise a way to investigate the structure of this object to make something of it. The approach we propose is to study the trajectories of a great number of simple steepest or random descent walks and, if statistical significance holds, figure out some properties of the landscape.

We run 10^4 SDW and RDW starting from points sampled at random and we examine the length of the walks, the cost of the local optima that are reached, the distance between each others. We use instances from the Tsplib [28] for which we know the shortest tour, att48, kroA100, tsp225. Though these instances are very few and of small size, the results obtained here are in accord with what we observed on many other instances of larger size while running the algorithm described below (see Section 18.3). The number of runs (10^4) was set in order to get some confidence in their statistical significance. We obtain the following conclusions:

1. starting from a tour t_{init}, the length of the walk is nearly equal to $\delta(t_{\text{init}}, t_{opt})$, and the length of random descent walk is nearly four times longer (t_{opt} is the shortest tour as provided in the Tsplib). According to Johnson [14], the length of SDW scales with n. However, for the small instances we use, this does not contradict our observations.

2. all local optima are gathered in a somewhat restricted region of the search space. For all local optima that are reached t_{lo}, $\delta(t_{\text{lo}}, t_{opt}) < n/3$ (see

Fig. 18.1). Furthermore, the distance between the local optima themselves is small. As the number of tours that are at a certain distance from t_{opt} scales like $n^{\delta(t,t_{opt})}$, it is clear that the region of the space where $\delta(t, t_{opt}) < n/3$ only holds a minority of points of the search space, resulting in a high concentration of local optimum in this region. Such a concentration was dubbed as a "deep valley" by Kirkpatrick and Toulouse [19], or "massif central" by Kauffman [17]. This experimental result is is accord with [19, 3], as well as with Stadler and Schnabl conjecture that there exists a massif central in the 2ETSP [31].

3. the distribution of the quality of the local optima that are reached by either the SDW and the RDW are identical. The distribution for local optima obtained by multiple runs of the RDW starting from the same initial random tour is also the same.

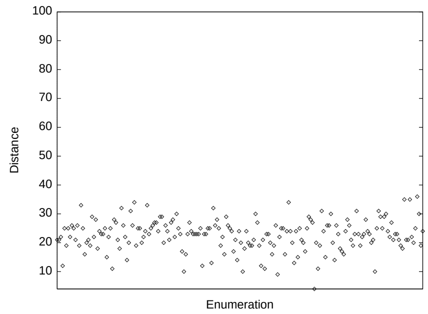
4. basins of attraction are highly intertwined: walks (either SDW, or RDW) starting from two neighboring points typically ends on two different local optima. From a visual point of view, basins of attraction should not be viewed as crater-like structures. Rather, they are more like very long and very thin canyons.

5. there is no correlation between the distance of the local optimum to the shortest tour and the length of the walk that finds it.

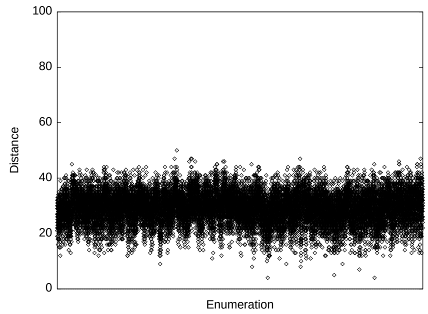
6. there is no correlation between the quality of the local optimum and the length of the walk that finds it.

Statistical/empirical studies such as these have severe drawbacks at first. Relying on intuition that can be faulty, it does not have the certainty of a mathematical proof. One may ask whether the big valley is a mirage or an actual fact. The fact that other authors have all obtained the same conclusion about the valley is on the positive side. Furthermore, to assess our conclusions, we had two distinct approaches. First, we designed instances having several well distinct, optimal tours. This distance between these shortest tours can be tuned from very small (the shortest tours are twin peaks in the landscape) to rather large (two shortest tours share at most 4 common edges), with all intermediate possibilities. Our intuition is that a big valley surrounds a shortest tour. As we expected, the local optima obtained by multiple runs of SDW either gathered in one region around the twin peaks, or they clearly gathered in different areas when the optimal tours are very distant from each others, or, in the intermediate case, the big valley gets wider and wider (the distribution of distances of couples of local optima shows that local optima are at increasing distances from each others, see Fig. 18.2.) Clearly, this all does not make impossible the case where an optimal tour would be left unfound. However, in the instances we used, the multiple runs did not find it. But they did find a "good" region.

A second approach relates to our original goal, the design of better heuristics. If using this knowledge about the structure of the search space we are able to



(a) Plot of the distribution of the distances to the optimal tour given in the Tsplib for the local optima that are found after SDW and RDW. In terms of distance, these local optima are very close to the best tour. The distance between any local optimum and the best tour is less than 1/3 of the size of the instance



(b) Plot of the distribution of the distance between any couple of local optima that are found after the 10^4 RDW. All local optima are concentrated in a small area. The distance between any two local optima is less than 2/3 of the size of the instance.

Figure 18.1 Massif Central of Local Optima. These figures were measured on kroA100. Note that the same results were obtained for every instance we measured.

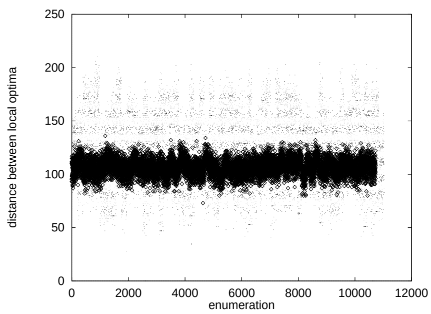


Figure 18.2 Plot of the distribution of the distance between any couple of local optima that are found after 200 RDW. The darker stripe, centered at y-coordinate 100, stands for RDW performed on instance lin318. The lighter cloud, spreading from y-coordinate 50 to 200, stands for RDW performed on a 300 cities instance with 2 global optima sharing only 4 edges.

design an algorithm that takes advantage of it to find local optima efficiently, this would also be a sign that this structure might be realistic.

Basically, we know that a short tour is to be found within a valley. Thus, the idea is to first find a valley with mere steepst descent, and then work inside the valley by recombining the local optima that have been found. At this stage, we can iterate the whole process to progress further inside the valley. We end-up with some kind of hybrid genetic algorithm. However, we have the advantage over standard approaches (see [27] for a review) to begin recombination in an area of the search space where it is really useful, instead of recombining tours of very low quality. This speeds-up our hybrid GA a lot as will be seen below.

Our approach has much to be compared with Martin et al.'s Chained Local Optimization (CLO) [23, 24] and Johnson's Iterated LK [14]. In the next section, we first present our hybrid GA, briefly review Johnson's version of ILK and Jünger et al.'s ILK, and provide some experimental results comparing the 3 algorithms. Note has to be taken that Johnson's ILK algorithm, together with Martin's CLO, are the best local search algorithms for the TSP to date, according to Johnson's review [14].

18.3 A HYBRID GA THAT SEARCHES THE TSP VALLEY

We have made our benchmarks with small populations of 10 individuals. Before all, the initial generation is optimized with 3-opt before attempting recombination. New individuals are then created by cross-over using a slightly refined version of the well-known enhanced-edge recombination (EER). We don't recall here the details of EER, and refer the reader to [34]. Our modification to EER relies on the fact that edges not belonging to any parents may have to be introduced in order to satisfy the tour constraint. It has been noted that this fact tend to decrease much the fitness of children. To try to limit the effect of this problem, we begin the construction of the tour with the shortest edge common to both parents (so we are never forced to replace it) and, when faced with the constraint, we use a ranked selection to choose the new edge with a probability inversely proportional to its length. Eventually the children are optimized with 3-opt. We use a "malthusian" replacement for the population : firstly we delete doublons, to keep diversity as the population size is small, then parents and children are sorted by fitness and the best individuals form the new population. We have chosen to limit the number of generations in such a way that the number of 3-opt calls is equal to the instance size, in order to compare our results with Johnson's ILK performances given in [14]. Results are shown in the GA column of Tab. 18.3 : average percent excess is given relatively to the optimum values given in the TSPLIB, and is the average computed on at least 10 runs. Results for 3-opt and Jünger et al.'s ILK comes from [16].

Running time are ranging from 40 s. to 10 mn. on Pentium 200 processor. These running times are longer than those of Johnson, but shorter than those of Jünger et al. Clearly, there is still room for improvement when compared to the top of the art heuristic from Johnson. Nonetheless, notice that the performances of our algorithm are rather less sensitive to the increase of instance

instance	Average percent excess			
	3-opt	Jünger et al.'s ILK	**GA**	Johnson's ILK
lin318	2.67	0.53	**0.41**	0.03
pcb442	3.01	0.90	**0.58**	0.03
u724	4.20	0.67	**0.49**	n.a.
pr1002	3.80	1.51	**0.68**	0.12

size than the others. Jünger et al.'s ILK consists in repeating successive independant runs of ILK, then selecting the best result. Johnson's implementation can be roughly described by saying that the result of a run is altered by a 4-opt random move and becomes the new starting solution for the next ILK local search. We think the difference in performance is explained by staying in the big valley (Johnson's and our algorithms), rather than starting again from bad solutions and being trapped by local optima at the border of the valley (Jünger et al.'s ILK) : this gives experimental evidence that a simpler 3-opt, well-used, can beat ILK which is much harder to implement.

18.4 CONCLUSION AND PERSPECTIVES

We have investigated the structure of the search space of instances of planar euclidian traveling salesman problem using 2-opt and 3-opt operators. This study confirms Kirkpatrick and Toulouse study with regards to the existence of a "deep valley" where a large number of local optima are gathered around (a) shortest tour(s). From that point, we do not imply that all local optima are concentrated in the valley. We merely mean that this structure is meaningful for steepest descent algorithms, thus metaheuristics such as tabu search and simulated annealing. Though very minute with regards to the overall search space, the valley still contains a large number of points. This valley is very attractive for steepest descent walks. There might be more than one valley when there are global optima very disimilar from each others. In this case, steepest-descent walks are attracted by one or the others. Basins of attraction seems to be highly intertwined, giving a canyon-like structure to the landscape, rather than a crater-like structure.

A lot of work has still be be done to have a good description of the landscape. It would be very interesting to be able to have indicators to predict the structure of the landscape of a given instance.

Current work on the structure of the QAP seems to imply that the 2ETSP is an extreme case of that of the QAP.

We have used this intuitive knowledge about the landscape of the 2ETSP to design a new hybrid genetic algorithm. We think that recombination is misused in standard GA's. Instead of recombining solutions taken at random, which are typically very bad solutions in the case of 2ETSP, it is much more efficient to first obtain good solutions, and only then, recombine them. These good solutions are obtained via steepest-descent walks started with random points, based on 2, or 3-opt. The local optima that are obtained are then recombined. The key idea is that we first come into the local optima valley and then, the recombination works inside the valley. The new points are optimized again via steepest descent walks giving new, better, local optima that can be recombined again, and so on. After a series of iterations, no improvement is made any longer the algorithm keeps on generating the same local optima at each iteration. This algorithm is very efficient and fast. Instances with many hundreds cities are solved to optimality within a few seconds on standard workstations. Its performance are comparable to that of Martin et al.'s chained local optimization and Johnson's ILK, the two top of the art heuristics for the TSP. A lot of work has still to be done to enhance our algorithm. For example, ILK uses LK while we use 2-opt, or 3-opt; the recombination operator is not very sophisticated: recombining more than two solutions seems attractive; more work on the implementation is needed to tackle instances of many tens of thousands cities. Despite the current limitations due to implementation, we think that our algorithm is promising for the future. Furthermore, based on an analysis of the structure of the problem search space, we think we have used an approach that can be brought to other problems. Some results on the QAP have already been obtained along this way.

References

[1] E. Aarts, J.K. Lenstra (eds), *Local Search in Combinatorial Optimization*, Wiley, 1997

[2] K.D. Boese, *Models For Iterative Global Optimization*, PhD Dissertation, UCLA, USA, 1996

[3] K.D. Boese, A.B. Kahng, S. Muddu, A New Adaptive Multi-Start Technique For Combinatoiral Global Optimizations, *Operations Research Letter*, **16**, 101–113, 1994

[4] G.A. Croes, A Method For Solving The Traveling Salesman Problems, *Operations Research*, **6**, 791–812, 1958

[5] M. Dorigo, L. Gambardella, Ant Colony System: A Cooperative learning approach to the traveling salesman problem, *Evolutionary Computation*, **1**(1), 1997

[6] J. Frank, P. Cheeseman, J. Stutz, When Gravity Fails: Local Search Topology, *Journal Of Artificial Intelligence Research*, **7**, 249–281, 1997

[7] M.R. Garey, D.S. Johnson, *Computers And Intractability: A Guide To The Theory Of NP-Completeness*, Freeman, 1978

[8] F. Glover, Tabu Search: part I, *ORSA Journal on Computing*, **1**, 190–206, 1989

[9] F. Glover, Tabu Search: part II, *ORSA Journal on Computing*, **2**, 4–32, 1990

[10] D. Goldberg, *Genetic algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989

[11] L.K. Grover, Local Search And The Local Structure Of NP-Complete Problems, *Operations Research Letter*, **12**, 235–243, 1992

[12] W. Hordijk, A Measure Of Landscapes, *Evolutionary Computation*, **4**(4), 1996

[13] A. Hertz, B. Jaumard, M. Poggi de Aragão, Local optima topology for the k-coloring problem, *Discrete Applied Mathematics*, **49**, 257–280, 1994

[14] D.S. Johnson, The Traveling Salesman Problem: A Case Study, in [1], 215–310, 1997

[15] D.S. Johnson, J.L. Bentley, L.A. McGeoch, E.E. Rothberg, *Near-Optimal Solutions To Very Large Traveling Salesman Problems*, to appear

[16] M. Jünger, G. Reinelt, G. Rinaldi, The Traveling Salesman Problem, *Network Models*, Handbooks in Operation Researchand Management Science, Volume 7, North-Holland Amsterdam, 225–330, 1995

[17] S. Kauffman, Adaptation on Rugged Fitness Landscapes, and Principles of Adaptation in Complex Systems, in D.L. Stein, *Lectures In The Sciences Of Complexity* volume I, Addison-Wesley, 1989

[18] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by Simulated Annealing, *Science*, **220**(4598), 671–680, 1983

[19] S. Kirkpatrick, G. Toulouse, Configuration Space Analysis Of The Traveling Salesman Problem, *Journal de Physique*, **46**, 1277–1292, 1985

[20] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (eds), *The Traveling Salesman Problem, A Guided Tour of Combinatorial Optimization*, Wiley, 1985

[21] S. Lin, Computer Solutions Of The Traveling Salesman Problem, *Bell System Technical Journal*, **44**, 2245–2269, 1965

[22] S. Lin, B. Kernighan, An Effective Heuristic Algorithm For The Traveling Salesman Problem, *Operations Research*, **21**, 498–516, 1973

[23] O. Martin, S.W. Otto, E.W. Felten, Large-step Markov Chains for the TSP Incorporating Local Search, *Operation Research Letter*, **1**, 219–224, 1992

[24] O. Martin, S.W. Otto, Combining Simulated Annealing With Local Search Heuristics, in G. Laporte, I.H. Osman (eds), *Metaheuristics in Combinatorial Optimization*, Annals of Operations Research, **63**, 57–75, 1996

[25] B. Manderick, M.D. Weger, P. Spiessens, The Genetic Algorithm and the Structure of the Fitness Landscape, in R. Belew (ed), *Proc. International Conf. on Genetic Algorithms*, Morgan-Kaufman, 1991

[26] Z. Michalewicz, *Genetic Algorithm + Data Structure = Evolution Program*, Springer-Verlag, 1995

[27] H. Mülhenbein, Genetic Algorithms, in [1], 137–171, 1997

[28] G. Reinelt, TSPLIB: A Traveling Salesman Problem Library, *ORSA Journal on Computing*, **3**, 376–384, 1991. The TSPLIB is available on the web.

[29] C. Reeves (ed), *Moden Heuristic Techniques for Combinatorial Optimization*, Blackwell, 1993

[30] P. Stadler, Towards A Theory Of Landscapes, in R. Lopès-Peña, R. Capovilla, R. Garcia-Pelayo, H. Waelbroeck, F. Zertuche (eds), *Complex Systems and Binary Networks*, Springer-Verlag, 1995

[31] P. Stadler, W. Schnabl, The Lansdscape Of The Traveling Salesman Problem, *Physics Letter A*, **161**, 337–344, 1992

[32] E.D. Weinberger, Correlated and Uncorrelated Fitness Landscapes And How To Tell The Difference, *Biological Cybernetics*, **63**, 325–336, 1995

[33] E.D. Weinberger, Local Properties of Kauffman's N-k model: A Tunably Rugged Energy Landscape, *Physical Review A*, **44**(10), 6399–6413, 1995

[34] D. Whitley, T. Starkweather, D. Shaner, Sceduling Problems and Traveling Salesman : the Genetic Edge Recombination Operator, *Proc. 3rd ICGA*, Morgan Kaufmann, 133–140, 1989 Chapter 22, 350–372, 1991

[35] M. Yannakakis, The Analysis Of Local Search Problems And Their Heuristics, in *Proc. STACS'90*, Lecture Notes In Computer Science, **415**, 298–311, 1990

[36] M. Yannakakis, Computational Complexity, in [1], 19–55, 1997