A Visual Method for Analysis and Comparison of Search Landscapes

Sebastian Volke
Leipzig University
Institut für Informatik
PF 100920, D-04009 Leipzig
volke@informatik.uni-leipzig.de

Dirk Zeckzer
Leipzig University
Institut für Informatik
PF 100920, D-04009 Leipzig
zeckzer@informatik.uni-leipzig.de

Gerik Scheuermann
Leipzig University
Institut für Informatik
PF 100920, D-04009 Leipzig
scheuermann@informatik.uni-leipzig.de

Martin Middendorf
Leipzig University
Institut für Informatik
PF 100920, D-04009 Leipzig
middendorf@informatik.uni-leipzig.de

ABSTRACT

Combinatorial optimization problems and corresponding (meta-)heuristics have received much attention in the literature. Especially, the structural or topological analysis of search landscapes is important for evaluating the applicability and the performance of search operators for a given problem. However, this analysis is often tedious and usually the focus is on one specific problem and only a few operators. We present a visual analysis method that can be applied to a wide variety of problems and search operators. The method is based on steepest descent walks and shortest distances in the search landscape. The visualization shows the search landscape as seen by the search algorithm. It supports the topological analysis as well as the comparison of search landscapes. We showcase the method by applying it to two different search operators on the TSP, the QAP, and the SMTTP. Our results show how differences between search operators manifest in the search landscapes and how conclusions about the suitability of the search operator for different optimizations can be drawn.

CCS Concepts

•Mathematics of computing → Optimization with randomized search heuristics; •Human-centered computing → Information visualization;

Keywords

Combinatorial optimization, Local search, Fitness landscapes, Empirical study

GECCO '15, July 11–15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3472-3/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754733

1. INTRODUCTION

A frequent class of optimization problems are *combinatorial* optimization problems, where an optimal solution has to be found from a finite set of solutions. Typically, the solution space grows exponentially in the problem size. Therefore, exhaustive search by complete enumeration is only possible for very small problem instances.

Local search algorithms [15] use search operators to construct a neighborhood among the solutions. They traverse the emerging search landscape in order to find locally optimal solutions. The topological complexity of the search landscape, e.g., indicated by the number of and the distance between local minima, the average length of search paths, or the probability of finding a good local minimum, depends on the interaction between the search operator and the optimization function. Optimization is much easier in search landscapes with simpler topology. Thus, an optimization problem can be solved by finding a search operator that induces a preferably simple search landscape.

The analysis of search landscapes is crucial in identifying well-performing search operators. In this paper, we present a method for analyzing search landscapes that is based on approximating the search landscape using random samples and steepest descent walks together with a visualization system that shows the topology of the solutions found by the sampling process. The method does not depend on a specific problem and thus is widely applicable. The visualization system fosters the interpretation of the analysis results. In particular, the visualization of the search landscape shows how the search algorithms perceives the optimization problem. We apply the method to instances of the TSP, the QAP, and the SMTTP and compare the performance of two different search operators on these problems.

2. SEARCH LANDSCAPES

2.1 Definition

Discrete combinatorial optimization addresses the selection of the best solution out of a finite *set of solutions* X of a problem instance. Usually, a *cost function* $f : X \to \mathrm{R}$

is used to model the quality of solutions: $x \in X$ is better than $y \in X$ if $f(x) < f(y)$. Solving the optimization problem is equivalent to finding the global minimum of f. Here, we are particularly interested in permutation problems, i.e., optimization problems where the set of solutions is a set of permutations of a given size, or a subset thereof. In the following, the length n of a permutation is referred to as the *problem size*. Many practically relevant permutation problems are NP-hard [9]. The relevant problems for this paper are discussed in Section 2.3.

Local search methods incrementally improve initial solutions until no further improvement can be achieved, i.e., until a local minimum is reached. Thereby, the *neighborhood* defines the search candidates for every solution. Formally, we define a neighborhood as a mapping $N : X \to 2^X$ that associates every solution with a set of neighbor solutions. In practice, the neighborhood is often implemented by use of a *search operator*. Following Schiavinotto et al. [24], we define a search operator Δ as a collection of operator functions $\delta : X \to X$ such that $y \in N(x) \iff \exists \delta \in \Delta$ with $\delta(x) = y$. The set of solutions together with the search operator form the *neighborhood graph* $G_{X,N} = (X, E_N)$. Thereby, the set of directed edges is defined as $E_N = \{(x, y)|x, y \in X \wedge y \in N(x)\} = \{(x, y)|x, y \in X \wedge \exists\delta \in \Delta : y = \delta(x)\}$. In particular, we consider search operators that give rise to connected and symmetric neighborhood graphs. Because the vertex set is the set of solutions, the cost function f can be considered a function on the vertex set of the neighborhood graph. Thus, the *search landscape* $S_{f,X,N} = (G_{X,N}, f)$ is formed.

The topological structure of the search landscape characterizes the difficulty of the search. Interesting properties are the number of local minima, the number of paths leading to these minima, or the lengths of these paths. The cost function f has an important influence on the complexity of the topology. Some problems give rise to much more complex search landscapes than others. However, also the search operator determines the topology, as it defines the connectivity within the landscape. Thus, complexity may also arise when the search operator does not fit the optimization problem well.

Depending of the size of the search operator—the operators defined in [24] have $|\Delta| \in O(n)$ or $|\Delta| \in O(n^2)$—the search landscape has $O(n!)$ nodes and $O(n^p n!)$ edges. This makes it impossible to search the complete graph except for very small n. Therefore, analysis approaches have been developed that use only parts of the landscape. One approach generates a representative sampling of the landscapes [17], i.e., it extracts a subset of solution with the same overall properties as the whole set of solutions. Other approaches consider search paths (steepest descent paths and descending random walks) and analyze their properties [8, 22, 26]. A *path* $\mathbf{p}$ *of length* l is a sequence of solutions $\mathbf{p} = [p_1 p_2 \dots p_l]$ with $p_{i+1} \in N(p_i)$ for all $1 \le i < l$. A *descending path* is a path $\mathbf{p}$ with $f(p_{i+1}) < f(p_i)$. Naturally, descending paths can only be extended as long as a better neighbor exists. In particular, descending paths end in local minima. For *steepest descent paths* we require additionally that $p_{i+1} = \min_{f(p)}\{p \in N(p_i)\}$. Of special interest are shortest paths within the landscape, especially their length. We define the distance $d_\Delta(x, y)$ between $x, y \in X$ as the length of a shortest path between x and y, i.e., as the minimal number of applications of operator functions $\delta \in \Delta$ that are necessary to transform x into y. For many operators, there exist efficient algorithms to compute the distance between two solutions or at least a good approximation for this distance [24].

2.2 Proximate Optimality Principle

The *Proximate Optimality Principle* (POP) states, that "good solutions at one level are likely to be found 'close to' good solutions at an adjacent level" [7, 11]. In the case of local search algorithms, it can be translated as such: good local minima are likely to be found "close to" each other. Thereby, closeness or proximity can be expressed by means of the search operator distance.

This formulation of the POP does by no means hold for every search landscape. If it holds, the result will be one or several clear accumulations of local minima within a small part of the search landscape. If this is the case, like for the Traveling Salesman Problem [8, 26], it is very beneficial for the search process and can be exploited. Therefore, we are particularly interested in investigating whether the POP holds for a search operator on given problems.

2.3 Optimization Problems

In the following, we define the three NP-complete optimization problems that we consider in this paper. The *Symmetric Traveling Salesman Problem* (STSP) is to find for n given locations and distances $d_{ij} = d_{ji}$ between locations $1 \le i, j \le n$ the shortest round-trip that visits all locations. A round-trip can be considered as a permutation π where $\pi(i)$ is the location at position i of the round-trip. The length of the tour is $f_{TSP}(\pi) = \sum_{i=1}^{n-1} d_{\pi(i),\pi(i+1)} + d_{\pi(n),\pi(1)}$.

The *Quadratic Assignment Problem* (QAP) requires n facilities to be assigned to n locations. Given are distances $d_{ij} = d_{ji}$ between two locations $1 \le i, j \le n$ and the amount of exchange $e_{kl} = e_{lk}$ between two facilities $1 \le k, l \le n$. An assignment of facilities to locations is a permutation π, where facility k is assigned to location $\pi(k)$. Then a permutation is searched that minimizes $f_{QAP}(\pi) = \sum_{k=1}^{n} \sum_{l=1}^{n} e_{kl} \cdot d_{\pi(k),\pi(l)}$.

The *Single Machine Total Tardiness Problem* (SMTTP) is to find an optimal, sequential schedule of n jobs. Each job i has a processing time $p_i > 0$ and a due date $d_i > 0$. A schedule is a permutation π where job i is scheduled at position $\pi(i)$. Then, the completion time of job i is $C_i(\pi) = \sum_{\pi(j) \le \pi(i)} p_j$. The tardiness of job i is $T_i(\pi) = \max\{0, C_i(\pi) - d_i\}$. A schedule π is searched that minimizes $f_{SMTTP}(\pi) = \sum_{i=1}^{n} T_i$.

2.4 Search Operators

The following two operators are of particular interest and will server as examples throughout the paper. The *2-opt operator* Δ_{2opt}, also known as 2-edge exchange operator, is a well established search operator for TSP ([5, 16]). Basically, it disentangles a route by eliminating edge crossings. Formally, the application of this operator corresponds to the reversal of a subsequence of a (circular) permutation. The corresponding distance between to permutations is called reversal distance. Unfortunately, the decision version of the problem to determine the reversal distance (or a corresponding shortest transformation between two solutions—also called the *sorting by reversal* problem) is NP-hard. However, the *bond distance* provides a good approximation for it: the number of edges is counted that are not in common between the two tours. This yields a maximal possible bond distance of n. The bond distance differs from the reversal distance at most by factor 2 as proven by Boese [2].

The *interchange operator* $\Delta_X = \{\delta_X^{ij} | 1 \leq i < j \leq n\}$ is the set of transpositions

$$\delta_X^{ij}(\pi) = (\pi_1 \dots \pi_{i-1}\pi_j\pi_{i+1} \dots \pi_{j-1}\pi_i\pi_{j+1} \dots \pi_n)$$

for a permutation $\pi = (\pi_1 \dots \pi_n)$. The distance between two permutations with respect to Δ_X can be determined by $d_{\Delta_X}(\pi, \pi') = n - c(\pi^{-1} \circ \pi')$ (see [24] for details). Thereby, $c(\pi)$ is the number of cycles of the permutation π, and π^{-1} is the inverse permutation, i.e., $\pi \circ \pi^{-1} = identity$. A permutation has at least one cycle, so that the maximal interchange distance is $n - 1$.

3. RELATED WORK

The analysis of search landscapes is an active area of research, e.g., [7, 8, 10, 15, 21, 22, 26]. Stadler and Schnabl [26] analyzed the auto-correlation function along random walks to gain insight into the shape of the landscape. Fonlupt et al. [8] also analyzed random walks as well as steepest descent walks, but mostly regarding their lengths. Primarily, they investigated the distribution of operator distances between local minima of the search landscape. Both of these papers focus on the TSP and also restrict their analysis to the 2opt operator. We consider our work a generalization of these approaches in that we extend their method and also expand the scope of the investigation to other optimization problems and other search operators. Additionally, we complement our method with a visualization that is tailored towards the analysis of the results.

There exist several sampling-based approaches for analyzing search landscapes since full enumeration is often infeasible because of the size of the landscape. Random sampling provides uninformed samples with unknown distribution within the landscape. Khor [17] presents a method based on Wang-Landau sampling that captures a representative subset of the search landscape. In particular, these subsets allow for reasoning about the distribution of cost values within the landscape. However, in this paper our main concern is the connectivity and relation among the local minima. We achieve this by considering a number of local search paths rather than unconnected solutions throughout the search landscape.

Flamm et al. [6] published an important work about the topological structure of landscapes and introduced the barrier tree. This work has been used in the analysis and visualization of combinatorial optimization [1, 14, 27, 28]. However, all of these methods require the set of all solutions below a certain cost threshold to be available. Therefore, they can only be applied to search landscapes of relatively small problems instances (see Section 2). Furthermore, no branch-and-bound techniques (as used by Hallam and Prügel-Bennett [14]) that would allow for efficient reduction of the search space are available for permutation problems in general. Our proposed approach does not result in a rigorous representation of the topology of the search landscape, but is applicable to problem instances that cannot be exhaustively searched. Instead of a tree representation of the search landscape, we derive visualizations that allow to draw conclusions about the shape of the search landscape from a small subset of it. A very similar approach has been presented by McCandlish [20] for the visualization of evolutionary landscapes. He uses an approximation of the number of evolutionary generations between individuals to obtain a transition or distance matrix. Afterward, he does an Eigendecomposition of this matrix to project the individuals into 2D space while preserving their evolutionary distance.

Halim et al. [13] propose a visualization system called "Viz" that facilitates the analysis of local search behavior. They create a search landscape representation by means of a 2D embedding of important landmarks from searches in the landscape, based on the Hamming distance function between the landmarks. Then, search runs are plotted as trajectories in this depiction. The focus of the system is the analysis of search algorithms, not on the search landscape itself. Furthermore, the fixed distance function introduces a gap between the perceived search space and the actual behavior of the search strategy. In contrast, we use distance functions that match the search operator in order to visualize the search landscape as seen by the search strategy.

4. ANALYSIS METHOD

The goal of our approach is to enable the topological analysis of search landscapes. The approach should be applicable to problem instances of arbitrary size. Further, it should not be specific to a certain problem type or search operator. Thus, the comparison of search landscapes of different problem instances, different search operators, and even different problems should be enabled. Thereby, the type of the problem should not need to be restricted. In particular, we do not rely on the existence of efficiently enumerable solution subsets or branch-and-bound techniques.

Our approach consists of two parts. In Section 4.1, the sampling of the search landscape providing representative data is described. The topological visualization (Section 4.2) uses this data enabling to address the goals and to perform the tasks in an effective and efficient manner.

4.1 Sampling of the Search Landscape

Analyzing the complete search landscape is infeasible for all but small problem instances (up to a problem size of about 15 for permutation problems). Instead, we analyze the properties of chosen solutions as well as of walks within the search landscape. Our analysis approach follows closely the approach of Fonlupt et al. [8, 22] and to some extend the one of Stadler and Schnabl [26].

We obtain a set of initial solutions, called *samples*, by generating random permutations using Knuth shuffles [18]. Then, we obtain a number of *search paths* by tracing each sample to a local minimum. Thereby, we use steepest descent paths that associate every solution with exactly one local minimum. The length of the search path provides information about the closeness of the minima to the samples.

We also obtain two additional sets of solutions from the search paths. First, we have the set of found *local minima*. Since multiple steepest descent paths may end at the same local minimum, the number of found local minima is equal to or less than the number of samples.

Second, we have the solutions on the search paths between the samples and the local minima. Since the lengths of search paths scale with the problem size [8], this can be a large amount of solutions. Adding all of them to the visualization would results in visual clutter. Therefore, we represent each path by its *median solution*.

We obtain information about the connectivity and the adjacency within the search landscape by computing the dis-

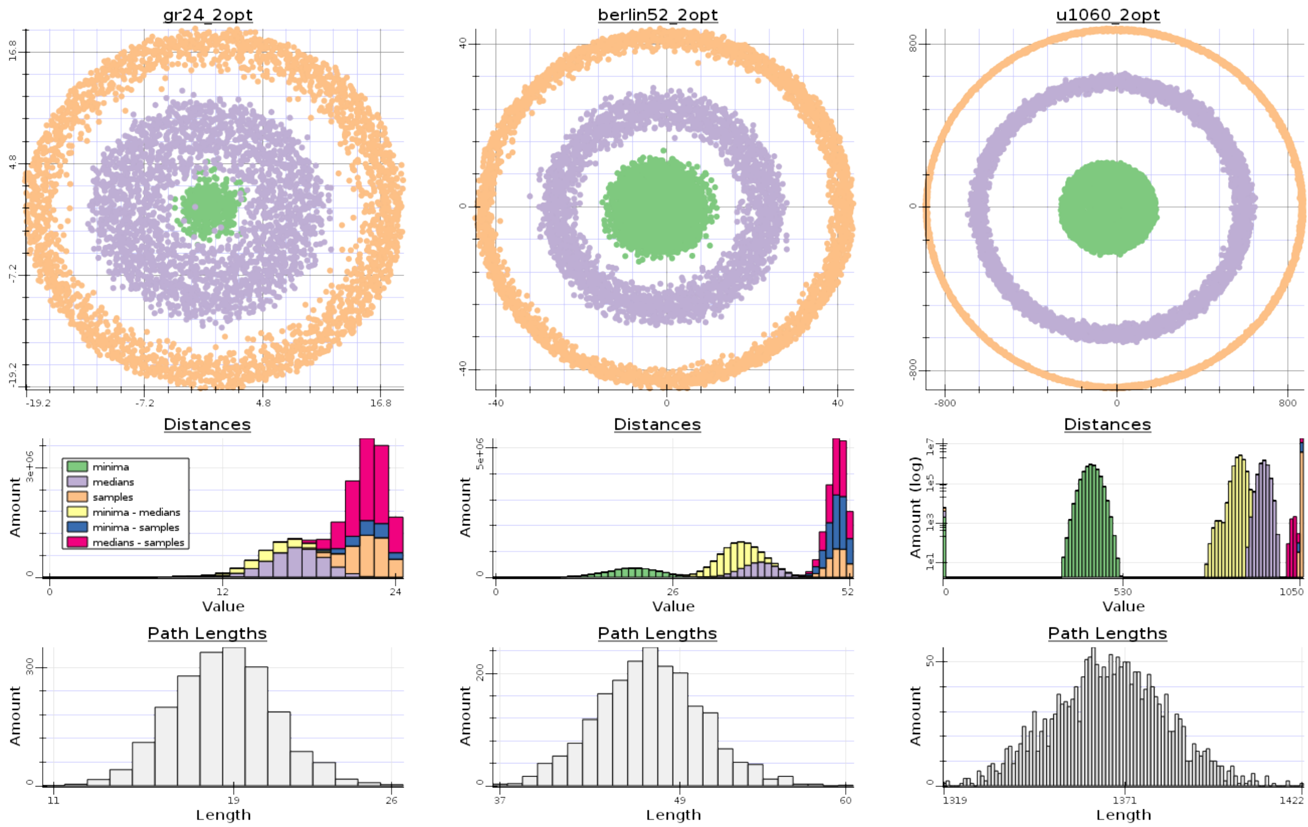

Figure 1: 2-opt search landscapes for three TSP instances from the TSPlib. The number in the problem names indicates the problem size. In the top row, the layout of the search landscapes is shown. Minima are shown in green, medians in light blue, and the samples (initial solutions) in orange. A clear, crater-like appearance can be seen. In the middle row, the distribution of the distances is shown. Colors correspond to the top row. The short distances between the local minima are clearly visible. In the bottom row, the distribution of the path lengths is plotted.

tance with respect to the search operator between any two solutions from all three sets (i.e., the samples, the corresponding local minima, and the corresponding median solutions). This results in six groups of distances. Based on knowledge about the search operator (e.g., the maximal distance or diameter of the search landscape, and the global distribution of distances), conclusions about the location and mutual proximity of local minima within the search landscape can be drawn [8]. Furthermore, the length of the search paths can be contrasted against the theoretical diameter of the search landscape.

Discussion.

Arguably, Knuth shuffles do not generate samples that represent the search landscape well in general [17]. During our experiments we found that only a small range of cost values is found by this sampling method and local minima or solutions near local minima are rarely encountered. However, random permutations are often used to initialize local search or simulated annealing methods. Therefore, our sampling method is in agreement with common optimization techniques. Because of that, we did not incorporate more elaborate sampling methods into our analysis.

Some analysis approaches (e.g., Stadler and Schnabl [26]) use random descending walks rather then steepest descent paths. These can be computed faster, because the neighborhood of each solution along the path does not need to be expanded completely. However, random walks typically can map each solution to several local minima. It is more straightforward to create a topological partition of the search landscape into basins around local minima using steepest descents [6] than using random descending walks [25]. There is also a close connection between steepest descent and the barrier tree [6]. Therefore, we preferred to use steepest descent in our method.

4.2 Visualization of the Search Landscape

In the following, we outline a visualization system that facilitates the understanding of the search landscape based on the extracted solutions (Section 4.1). For every solution we have a permutation and the associated cost value. Additionally, the distance between any two solutions is known. We propose using (stacked) bar charts of histograms [30] and scatterplots [12] for presenting the data. A detailed discussion of design alternatives and the design choices made are presented in a companion paper [29].

The distribution of the paths' length is shown using bar charts (e.g., bottom row of Figure 1). Histograms are very well suited for the analysis of single attributes.Thereby, the range of possible values is partitioned into equally-sized bins

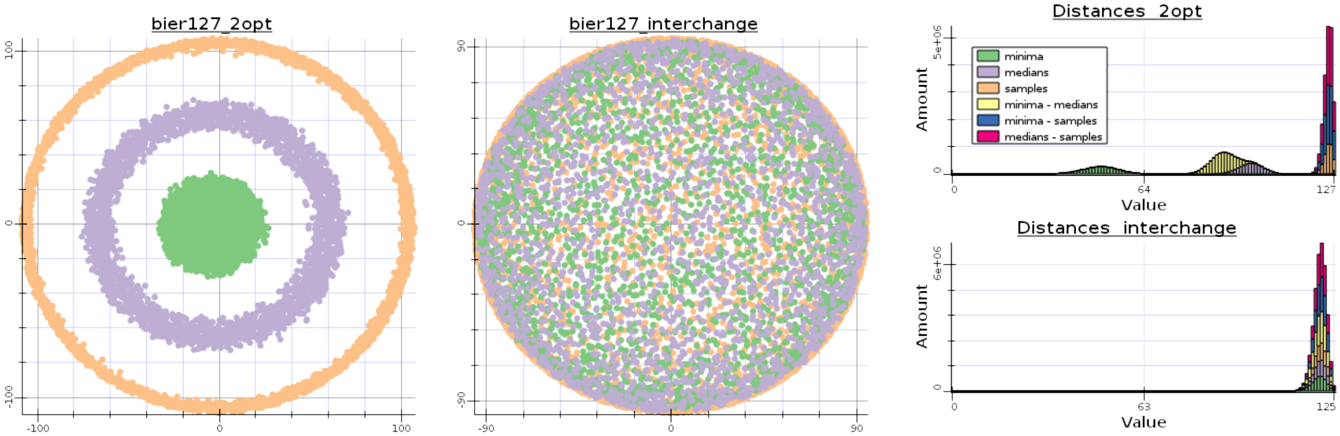

Figure 2: Juxtaposition of the layouts (left image and center image) and the distance distributions (right images) of the search landscape of the Bier127 problem for the 2-opt operator and the interchange operator. The search landscape from the interchange operator reveals much less structure, and the distances within all groups of solutions are almost maximal.

and the number of data values in every bin is counted. The bins are represented by bars in a two-dimensional plot. The position of the bar on the x-axis corresponds to the range of the bin, while the height of the bar (y-axis) corresponds to the number of elements in the bin. Altogether, this results in a so-called bar chart.

The distances between the solutions are represented using stacked bar charts and scatterplots. In the stacked bar chart (e.g., the middle row of Figure 1), six groups of distances are shown: distances within one class of solutions (3), and distances between two of these classes (3). For each group of distances, its histogram is computed. Stacking the histograms then yields the stacked bar chart. Thereby, each group of distances is assigned a particular color, e.g., green for the group of distances between minima, and yellow for the group of distances between minima and median solutions. The color assignment is shown in a legend.

For showing the distances, the stacked bar charts are complemented by a visual depiction of the search landscape (e.g., top row of Figure 1). Therefore, we exploit that the distances between the solutions carry topological information. The constructed visualization should preserve the distances and the topological properties as good as possible. Thereby, the topological proximity is approximated by the spatial proximity in the plot. To do so, the solutions are mapped onto the 2D plane such that the Euclidean distances between the points in the plane approximates the distances within the search landscape. This mapping is achieved using a metric Multidimensional Scaling technique [4], in particular, a variant of the Shepard-Kruskal algorithm that simulates a spring force model is applied. Initially, all solutions are placed at random positions in the 2D plane. Each solution exerts a force on every other solution according to the deviation of the Euclidean distance to the operator distance (for details see [29]). Then, all solutions are moved, so that the overall force is minimized. This is repeated until an equilibrium or a maximal number of iterations is reached. The resulting layout is shown in a 2D scatterplot. The same colors assigned to the solution classes for depicting the stacked bar charts are used for coloring the points in the scatterplot.

5. RESULTS

The search landscapes obtained by applying the 2opt and the interchange operator to TSP, QAP, and SMTTP were analyzed. The results of this analysis are presented next.

5.1 TSP

The TSP is a classical benchmark problem for which well-established results are available [8, 26]. Analyzing the search landscapes of the 2opt operator on all instances of the TSPlib [23] up to a problem size 1048 using our method confirmed these results. Figure 1 shows the result for three example problems of different sizes. Similar images have been obtained for all other test instances. The local minima (green) form a cluster in the center of the search landscape layout (Figure 1, top row). From the stacked bar charts, we can see, that the distances between the local minima are $\frac{1}{3}$ to $\frac{1}{2}$ of the problem size (Figure 1, middle row). Given the definition of the bond distance (cf. Section 2.4), this means that each two local minima have $\frac{1}{2}$ to $\frac{2}{3}$ inter-city connections in common. This conformance is unexpectedly high. Furthermore, we can see that the distances between the samples (orange) are almost maximal and that there is a large distance between the samples and the local minima (dark blue). The average path length is slightly smaller than the diameter of the search landscape. This indicates, that 2opt converges relatively straight to local minima.

Further, we compared the search landscapes of two different search operators (2opt and interchange operator). Figure 2 shows a juxtaposition of both search landscapes for instance Bier127 from the TSPlib, which is representative for the other instances from the TSPlib. When using the interchange operator, the search paths are longer on average and the found local minima (green) are not located close each other. Instead, there are almost maximal distances both between the samples (orange) as well as between the local minima (green). This results in a very different landscape layout that reveals little structure.

5.2 QAP

We applied our method to all instances of the QAPlib [3]. Results for two instances are shown in Figure 3 as an ex-

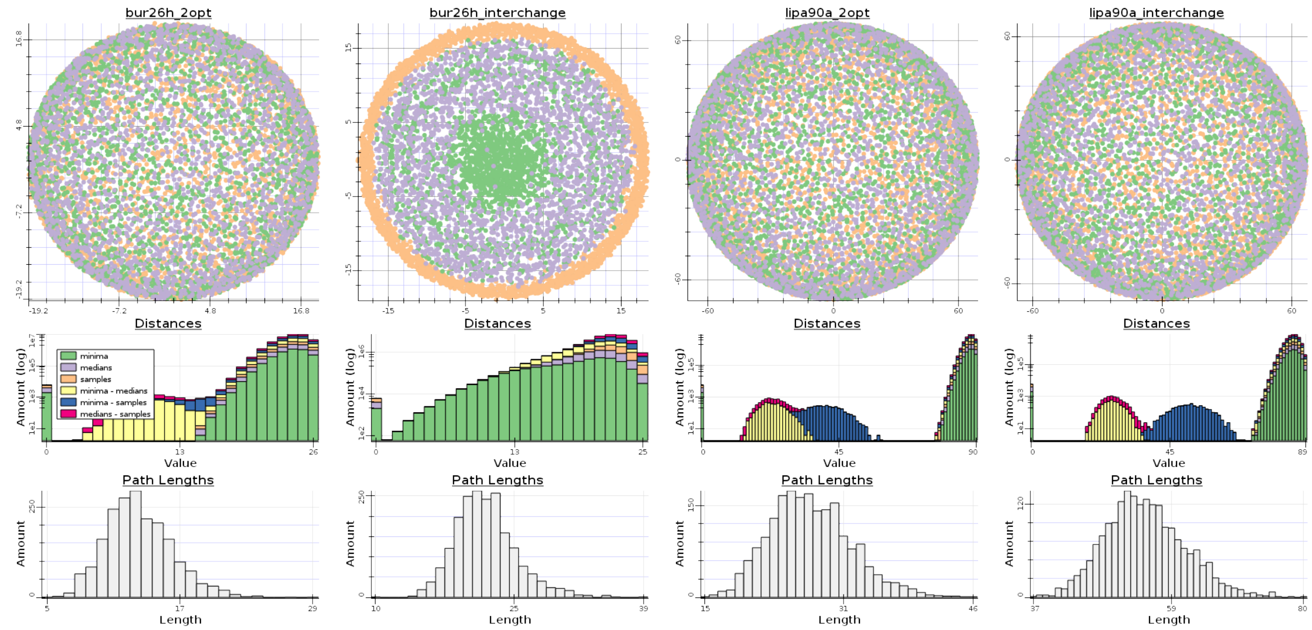

Figure 3: Search landscapes of two instances from the QAPlib for the 2opt operator and the interchange operator. The number in the problem names indicates the problem size. The search landscape layout is shown in the top row. Minima are shown in green, medians in light blue and the samples (initial solutions) in orange. In the middle row, the distribution of the distances is shown. Colors correspond to the top row. At the bottom, histograms of the search path length distributions are shown.

ample. Different from the TSP, we typically did not find the characteristic crater-like structure in the QAP. Usually, as can be seen in the search landscape layouts in Figure 3, top row, the local minima (green) are scattered throughout the landscape. Distances between solutions of the same group are nearly maximal almost without exception. In many problem instances, e.g., the lipa90a instance (see right images in Figure 3), there is no visual difference between the search landscapes of both operators. However, the interchange operator produces longer search paths (cf. path length histograms in the bottom row of Figure 3). The path length bar charts show that for some instances the local minima are close to the samples, less than half of the diameter of the search landscape (first, third, and fourth column of Figure 3). Interestingly, in some instances a subset of the found local minima (green) are grouped together when using the interchange operator (cf. the second column in Figure 3), whereas others remain scattered throughout the landscape. This can also be seen by the two peaks in the distribution of distances between local minima in the histogram. Such a grouping of local minima does not exist for the 2opt operator in any of the instances of the QAPlib.

5.3 SMTTP

For the analysis of the SMTTP, we generated various test instances of size 100. Both, the job lengths as well as the due dates were determined randomly. The processing times were chosen uniform randomly between 1 and 100. In the following, T denotes the sum of all processing times, i.e., the total processing time of the schedule. The due dates where generated according to two parameters r (range length) and $\widetilde{d}$ (mean due date). Thereby, the due dates were chosen uniformly from the interval $\left[T \cdot (\widetilde{d} - \frac{r}{2}), T \cdot (\widetilde{d} + \frac{r}{2})\right] \cap [0, T]$. The larger $\widetilde{d}$ is the earlier are the due dates, so that the total tardiness increases.

Figure 4 shows the search landscapes of the 2opt operator and the interchange operator for four selected instances. We found that the due date parameter $\widetilde{d}$ has a strong influence on the characteristics of the search landscape. If the mean due date is low, almost no structure is visible (top row). As can be seen in the bottom row of Figure 4, this changes when the mean due date increases. Then, a clear separation between local minima (green) and initial solutions (orange) can be seen, forming a crater-like structure similar as for the TSP with 2opt (cf. Figure 1). This structure is already visible in the search landscapes of the interchange operator at smaller due date values than in the landscapes of the 2opt operator. The range length appears to have an influence on the sharpness of the crater. Larger distances within the crater are possible when the range length is small.

6. DISCUSSION

We found the results of Fonlupt et al. [8] about 2opt on the TSP confirmed by our experiments. The "massif central" that is mentioned both by Fonlupt (and Stadler [26]) is clearly visible in the layout of the search landscapes. The distances of $n/3 - n/2$ between the local minima (cf. Figure 1) are in accordance with Fonlupt's findings as are the average lengths of the steepest descent paths. All this indicates that the POP holds for the TSP with the 2opt operator.

When applying the interchange operator to the TSP, we obtain longer search paths and the search landscape layouts reveal almost no structure. This confirms intuitively that

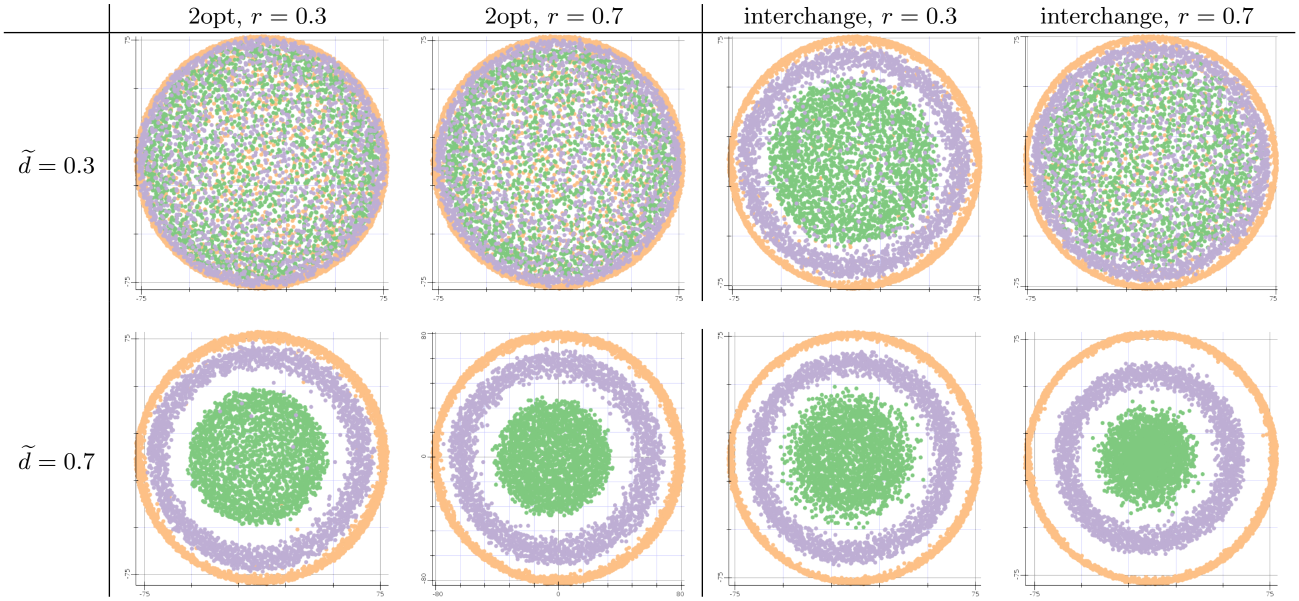

Figure 4: Randomly generated instances of the SMTTP with size 100. r is the relative length of the due dates interval and $\widetilde{d}$ is the mean due date. The layouts of the search landscapes of the 2opt operator and the interchange operator are shown. Local minima are shown in green, medians in light blue and the samples (initial solutions) in orange.

steepest descent using the 2-opt operator is more efficient than using the interchange operator [19, 26]. Our interpretation is that the missing structure makes the TSP harder when using the interchange operator. In any case, the POP does not hold for the interchange operator on the TSP.

In contrast to the TSP, we did not find much structure in the landscapes of the QAP with our method. Moreover, algorithms are prone to get stuck in local minima that are close to the sample points, which can be induced from the short length of the search paths compared to the distances of the minima and the diameter of the search landscape. The interchange operator leads to longer search paths in comparison to the 2opt operator. Furthermore, in some instances, at least some structure among the local minima was present in the landscapes of the interchange operator. From that we conclude that the interchange operator seems to fit the QAP better than the 2opt operator. However, even for the interchange operator there is no clear clear indication that the POP holds. This indicates that the QAP is a harder problem than the TSP.

The results in the previous section show that the difficulty of the SMTTP clearly depends on the mean due date. The later the mean due date, the higher the tardiness and the more crater-like the landscape. This indicates that the POP holds and that the problem is potentially easier to solve. A reason for this may be the asymmetry in the cost function of the SMTTP, because of the cut-off of negative tardiness. The higher the overall tardiness is, the smaller is the overall cut-off. Thus, changes in the schedule are reflected to a greater extend by changes in the cost function. Also—in contrast to the TSP—the crater-like structure is more pronounced for the interchange operator compared to the 2opt operator, especially for small mean due dates $\widetilde{d}$ and small ranges r. This indicates that for SMTTP, search algorithms based on the interchange operator might produce better results than those based on the 2opt operator.

7. CONCLUSIONS AND FUTURE WORK

We presented an analysis method for search landscapes that is applicable to permutation problems without restrictions, as well as to many different search operators. It uses steepest descent paths from random solutions within the search landscape to reveal topological properties of the landscape. The method is in accordance with and extends previous work on search landscapes [7, 8, 22, 26]. It includes a visualization system [29] based on visualization of statistical data as well as on a topological visualization of the search landscape. This supports the interpretation of the search landscape. We applied the method to instances of the TSP, the QAP, and the SMTTP. Our findings confirm previous work on the TSP and common assumptions about the relation between TSP, QAP, and SMTTP. Furthermore, we showed the influence of two due date parameters on the difficulty of the SMTTP and presented indications that the interchange operator is better suited for search algorithms for QAP and SMTTP than the 2opt operator.

Our findings suggest, that a more complete study that covers more search operators might reveal interesting insights into these well-known problems. In particular, a search operator that leads to crater-like structures in the QAP landscapes would be worth finding.

Sophisticated metaheuristics combine multiple search operators. Therefore, it is interesting future work to investigate how our approach can be extended to such cases.

8. ACKNOWLEDGMENTS

This work was partly supported by European Social Fund (ESF App.No.100098251).

9. REFERENCES

[1] S. Bin, S. Volke, G. Scheuermann, and M. Middendorf. Comparing the Optimization Behaviour of Heuristics with Topology Based Visualization. In A.-H. Dediu, M. Lozano, and C. Martín-Vide, editors, *Theory and Practice of Natural Computing*, volume 8890 of *Lecture Notes in Computer Science*, pages 47–58. Springer, 2014.

[2] K. D. Boese. *Models for Iterative Global Optimization*. PhD thesis, University of California at Los Angeles, CA, USA, 1996.

[3] R. Burkard, S. Karisch, and F. Rendl. QAPLIB – A Quadratic Assignment Problem Library. *Journal of Global Optimization*, 10(4):391–403, 1997.

[4] T. Cox and M. Cox. *Multidimensional Scaling*. CRC Press, 2010.

[5] G. A. Croes. A method for solving traveling salesman problems. *Operations Research*, 6:791–812, 1958.

[6] C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. Barrier Trees of Degenerate Landscapes. *Z. Phys. Chem.*, 216:1–19, 2002.

[7] C. Fleurent and F. Glover. Improved Constructive Multistart Strategies for the Quadratic Assignment Problem Using Adaptive Memory. *INFORMS Journal on Computing*, 11(2):198–204, Feb. 1999.

[8] C. Fonlupt, D. Robilliard, P. Preux, and E.-G. Talbi. Fitness Landscapes and Performance of Meta-Heuristics. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 257–268. Kluwer, 1999.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[10] J. Garnier and L. Kallel. How to Detect All Maxima of a Function. In *Theoretical Aspects of Evolutionary Computing*, pages 343–370. Springer, 2001.

[11] F. Glover and M. Laguna. *Tabu Search*. Kluwer, 1997.

[12] G. Grinstein, M. Trutschl, and U. Cvek. High-Dimensional Visualizations. In *Proceedings of the VII Data Mining Conference KDD Workshop 2001*, pages 7–19. ACM Press, 2001.

[13] S. Halim, R. H. C. Yap, and H. C. Lau. Viz: A Visual Analysis Suite for Explaining Local Search Behavior. In *User Interface Software and Technology*, pages 57–66, 2006.

[14] J. Hallam and A. Prügel-Bennett. Large Barrier Trees for Studying Search. *IEEE Trans. Evolutionary Computation*, 9(4):385–397, 2005.

[15] H. H. Hoos and T. Stützle. *Stochastic local search: Foundations & applications*. Elsevier, 2004.

[16] D. S. Johnson. Local optimization and the Traveling Salesman Problem. In M. S. Paterson, editor, *Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 446–461. Springer, 1990.

[17] S. Khor. Search space analysis with Wang-Landau sampling and slow adaptive walks. *CoRR*, abs/1112.5980, 2011.

[18] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley, 1997.

[19] S. Lin and B. W. Kernighan. An Effective Heuristic Algorithm for the Travelling-Salesman Problem. *Operations Research*, 21:498–516, 1973.

[20] D. M. McCandlish. Visualizing fitness landscapes. *Evolution*, 65(6):1544–1558, 2011.

[21] E. Pitzer and M. Affenzeller. A Comprehensive Survey on Fitness Landscape Analysis. In J. Fodor, R. Klempous, and C. Suárez Araujo, editors, *Recent Advances in Intelligent Engineering Systems*, volume 378 of *Studies in Computational Intelligence*, pages 161–191. Springer, 2012.

[22] P. Preux, D. Robilliard, C. Fonlupt, R. M. Karp, and J. M. Steele. Fitness Landscapes of Combinatorial Problems and the Performance of Search Algorithms, 1997.

[23] G. Reinelt. TSPLIB—A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[24] T. Schiavinotto and T. Stützle. A Review of Metrics on Permutations for Search Landscape Analysis. *Computer and Operations Research*, 34(10):3143–3153, October 2007.

[25] B. Stadler and P. Stadler. Combinatorial vector fields and the valley structure of fitness landscapes. *Journal of Mathematical Biology*, 61(6):877–898, 2010.

[26] P. F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161(4):337 – 344, 1992.

[27] S. Volke, S. Bin, D. Zeckzer, M. Middendorf, and G. Scheuermann. Visual analysis of discrete particle swarm optimization using fitness landscapes. In H. Richter and A. Engelbrecht, editors, *Recent Advances in the Theory and Application of Fitness Landscapes*, volume 6 of *Emergence, Complexity and Computation*, pages 487–507. Springer, 2014.

[28] S. Volke, M. Middendorf, M. Hlawitschka, J. Kasten, D. Zeckzer, and G. Scheuermann. dPSO-Vis: Topology-based Visualization of Discrete Particle Swarm Optimization. *Computer Graphics Forum*, 32(3):351–360, 2013.

[29] S. Volke, D. Zeckzer, M. Middendorf, and G. Scheuermann. Visualizing topological properties of the search landscape of combinatorial optimization problems. Accepted for presentation at *Topology-Based Methods in Visualization 2015* (Annweiler, Germany, May 20. – 22., 2015). Final version for workshop proceedings currently under review.

[30] L. Wilkinson. *The Grammar of Graphics*. Springer, 1999.