Coarse-Graining Large Search Landscapes Using Massive Edge Collapse

Sebastian Volke, Martin Middendorf, and Gerik Scheuermann

Abstract A thorough understanding of discrete optimization problem instances is the foundation for the development of successful solving strategies. For this, the analysis of search spaces is a valuable tool. In particular, networks of solutions—referred to as *search landscapes*—are used in research. Because of the large number of solutions, topological analysis methods are typically restricted to much smaller problem instances than instances that occur in practical applications. In this paper we present a coarse-grained abstraction of search landscapes—the meta landscape—that is accessible for a complete analysis, can be computed easily and preserves relevant properties of the original search landscapes. Thus, detailed topological analysis and visualization become available for problem instances of realistic sizes. We demonstrate the use of our method for search spaces of more than 10^{50} solutions.

Keywords Combinatorial optimization · Fitness landscapes · Topological analysis · Barrier trees

1 Introduction

Combinatorial optimization is a challenging task. One reason is the large search space of typical optimization problems. Therefore, a better understanding of large search spaces is of high interest, but difficult to obtain. It would be helpful if humans could use their visual pattern detection abilities to derive structural properties of search spaces. The knowledge about such properties can then be used to design good (meta-)heuristics for the problem. However, such a visual inspection requires a visualization of the search space. Obviously, any direct visualization of all solutions

S. Volke (✉) · M. Middendorf · G. Scheuermann
Institut für Informatik, Leipzig University, Leipzig, Germany
e-mail: volke@informatik.uni-leipzig.de; middendorf@informatik.uni-leipzig.de; scheuermann@informatik.uni-leipzig.de

H. Carr et al. (eds.), *Topological Methods in Data Analysis and Visualization V*, Mathematics and Visualization, https://doi.org/10.1007/978-3-030-43036-8_4

is usually impossible since even small problems, e.g., the Traveling Salesman Problem with $n \geq 13$ cities, have more solutions than there are pixels on any screen.

Research has found ways of obtaining a structure-preserving, coarse representation of the search space. In this paper we present a novel approach that relies on a massive and implicit collapse of parts of the search space. The method exploits structural properties of the solution set and the chosen search operator, but is independent of the cost function of the optimization problem. The solutions are partitioned into a manageable number of well-describable subsets. On top of them, the coarse-grained *meta landscape* is constructed that approximates the original search space. The meta landscape is small enough such that complete (topological) analyses are feasible. This enables the use of established visualization techniques like dPSO-Vis [18] for the investigation of the search space.

We present a precise mathematical definition of the meta landscape and discuss to what extent topological properties of the meta landscape correspond to topological properties of the original search space. We further discuss how approximate meta landscapes can be constructed for problem instances that are too large for an exhaustive search. Finally, we present and discuss different heuristics for constructing approximate meta landscapes that reveal different aspects of the search space. For the examples of the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP), we show how the proposed method can be used in practice to analyze optimization problem instances.

2 Foundations and Related Work

In this section, the mathematical foundations of this work are introduced and related work is reviewed.

2.1 Discrete Optimization Problems

A *discrete optimization problem* (also called *combinatorial optimization problem*) is to find the best solution out of a finite *set of solutions* X where the quality of a solution is determined by a *cost function* $f : X \to \mathbb{R}$, so that $x \in X$ is better than $y \in X$ if $f(x) < f(y)$. Thus, solving a discrete optimization problem is equivalent to finding the global minimum of f. For many discrete optimization problems the set of solutions can be characterized easily, but its size is so large that a complete enumeration is infeasible even for small problem sizes. Many practically relevant discrete optimization problems are NP-hard [5].

Common types of discrete optimization problems are subset problems and permutation problems. Subset problems, like the maximum binary variable saturation problem (MAXSAT), are defined on the power set X over the finite set $\{1, \ldots, n\}$, or equivalently over the set of all binary strings of length n, i.e., $|X| = 2^n$. Permutation problems are defined on the set X of permutations of size n, i.e., $|X| = n!$. Examples are the *Traveling Salesman Problem* (TSP) and related vehicle routing problems, job scheduling problems and the *Quadratic Assignment Problem* (QAP) [5].

The techniques that are proposed in this paper are, in principle, applicable to any type of discrete optimization problem. However, we restrict our presentation to permutation problems and use the TSP and the QAP as particular examples.

The symmetric TSP is: Given a set of n cities and distances $d_{ij} = d_{ji}$ between cities $1 \leq i, j \leq n$, the task is to find the shortest round-trip that visits all cities exactly once. A round-trip can be modeled as a permutation π, where $\pi(i)$ is the city at the i-th position of the round-trip. Then, $f_{TSP}(\pi) = \sum_{i=1}^{n-1} d_{\pi(i),\pi(i+1)} + d_{\pi(n),\pi(1)}$ is the length of a tour. There is no distinguished "first" city, and since d_{ij} is symmetric, the direction of the round-trip is not important. Thus, there are $(n-1)!/2$ equivalence classes of solutions.

The QAP is: given are n facilities with a flow F_{ij} between facilities i and j and n locations with distances d_{ij}, $1 \leq i, j \leq n$. Then, a mapping π of the facilities to the cities is to be found, such that $f_{QAP}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} F_{ij} \cdot d_{\pi(i)\pi(j)}$ is minimized. The QAP does not exhibit any symmetries in general, so that $n!$ equivalence classes of permutations are needed to represent all possible solutions.

2.2 Search Landscapes

The notion of landscapes was first presented in 1932 by Sewall Wright [21] as *fitness landscapes* to model evolutionary processes. The concept has been used many times to study optimization [1, 6, 12, 16, 18, 19]. A (fitness) landscape $L(X, N, f)$ consists of a finite set X, a neighborhood relation $N \subset X \times X$, and a function $f : X \to \mathbb{R}$ [12]. Depending on the context, f is called fitness function, energy function, or cost function. N induces a graph structure on the set of nodes X. Thus, landscapes can also be defined by a pair $(G_{X,N}, f)$ for a function $f : X \to \mathbb{R}$ and graph $G_{X,N} = (X, N)$ with vertex set X and edge set N [3]. Landscapes are a valuable tool for the study of local search algorithms [12] where the *neighborhood* of the local search algorithm defines the edge set of the graph. In practice, the neighborhood is often modeled by means of a *search operator* Δ, that is a collection of operator functions $\delta : X \to X$ [15]. Then, $(x, y) \in N \iff \exists \delta \in \Delta$ such that $\delta(x) = y$.

For a combinatorial optimization problem with set of solutions X, cost function f, and a local search algorithm with neighbourhood N the corresponding *search landscape* is $L(X, N, f) = (G_{X,N}, f)$. In this paper we assume without loss of generality that the graph is connected and consider only symmetric neighbourhoods.

2.3 Landscape Analysis

Landscape analysis is used in different fields of research and various analysis methods have been developed (see [12] for a comprehensive survey). Of particular interest is the investigation of barriers between local minima. Together with the concept of basins of attraction, barriers can be used to structure a landscape topologically. This gives rise to the barrier tree that was described in detail by Flamm et al. [3]. It is a rooted tree that has the local minima and saddle points of the landscape as its nodes and the hierarchical relations between them are represented by the edges. A major drawback of the barrier tree is that it can only be applied if all local minima and all barriers are known. To find them it is often necessary to completely enumerate at least the part of the landscape below a particular barrier. Thus, the barrier tree is often limited to the analysis of smaller problem instances, e.g., TSP instances with up to 13 cities ($2.4 \cdot 10^8$ solutions).

A different approach for the analysis of search landscapes is the computation of the auto-correlation of random walks, as done in [16], and the investigation of distances between local minima, as done in [4, 13]. Both approaches indirectly reveal some structural properties of the search landscape of TSP instances. While not requiring a complete enumeration of all solutions, these approaches have the drawback that no comprehensive topological structure of the search landscape is determined. Only indications about the shape of the search landscape can be obtained. The methods have been applied to problems instances up to approximately 400 cities [4, 16].

Local Optima Networks (LON) [8, 10, 11] have been introduced to investigate search methods. The idea is to reduce the search space to a (sampled) selection of local minima. The connectivity is defined through means of a so called escape perturbation operator that mutates a local optimum in an attempt to escape from its basin of attraction. Being defined on minima, LONs do not provide context for arbitrary solutions in the search space. Although topological analyses are possible on LONs [7], the results only describe the behavior of the escape operator on the set of local optima in the search landscape with respect to the search operator.

In this work, we propose a method that allows to approximate the topological structure for search landscapes that cannot be enumerated completely. It applies to single-operator landscapes and incorporates all solutions of the search space.

2.4 Landscape Visualization

There exist direct visualization approaches that place individual landscape nodes in the Euclidean plane by using multidimensional scaling techniques. This can be based, e.g., on the probability of transitions between solutions [9], or on distances within the landscape [20]. Potential disadvantages of these approaches comprise the error inherent to projection techniques as well as the reliance on samples of the landscape.

There exist also topological visualization approaches. A graph layout of the barrier tree is used, e.g., by Hallam and Prügel-Bennet [6]. In this work, we build upon the visualization tool dPSO-Vis by Volke et al. [18, 19]. In the latter approach, a 1D height function is computed that has the same barrier tree as the original landscape. The focus of that work was the study of folding landscapes of RNA molecules, but the idea has also been applied to study search operators for the TSP [1].

3 Coarse-Grained Search Landscapes: The *Meta Landscape*

A main problem with search landscapes of combinatorial optimization is the exponential number of landscape nodes. This makes a topological analysis infeasible even for problem instances of small size. We tackle this problem by creating a coarse-grained abstraction of the search landscape that is amenable to complete computational analysis. In the following, we define the *meta landscape* that is the foundation for the coarse-graining. Then, some properties of meta landscapes and how their topology relates to the topology of the original search landscape are discussed.

3.1 Definition

Given a search landscape $L(X, N, f)$, we introduce the *meta landscape* $L(\tilde{\mathbf{X}}, \tilde{N}, \tilde{f})$. Basically, the approach can be regarded as a simplification of the search landscape by means of collapsing edges. Formally, a *meta solution* $\tilde{X} \subset X$ is defined as a set of solutions where the subgraph $G[\tilde{X}]$ of the neighborhood graph with vertex set $\tilde{X}$ is connected. A set of meta solutions $\tilde{\mathbf{X}} \subset 2^X$ *represents* X if $\tilde{\mathbf{X}}$ is a partition of X, i.e., the meta solutions are pairwise disjoint and together cover X: $\bigcup_{\tilde{X} \in \tilde{\mathbf{X}}} \tilde{X} = X$.

We define a neighborhood relation $\tilde{N}$ between meta solutions by $(\tilde{X}, \tilde{Y}) \in \tilde{N} \iff \exists x \in \tilde{X}, y \in \tilde{Y} : (x, y) \in N$. The cost function f is transferred to the meta solutions by constructing the function $\tilde{f} : \tilde{\mathbf{X}} \to \mathbb{R}, \tilde{X} \mapsto \min\{f(x) \mid x \in \tilde{X}\}$. Thus, every meta solution is represented by the local optimum that it contains. Now, $L(\tilde{X}, \tilde{N}, \tilde{f}) = (G_{\tilde{\mathbf{X}}, \tilde{N}}, \tilde{f})$ is called *meta landscape*. Observe, that the meta landscape is itself a search landscape, so that topological analysis approaches for search landscapes can be applied to meta landscapes as well.

3.2 Properties

Meta landscapes have some useful properties which indicate that meta landscapes can be considered a valid topological compression method for search landscapes. First, we note that for a maximal partition of X, i.e., $\forall \tilde{X} : |\tilde{X}| = 1$, the meta landscape is equivalent to the original search landscape and thus possesses the same barrier tree. Hence, the approximation through the meta landscape converges towards exactness with the degree of its granularity. If the partition is coarse-grained, the meta landscape conceals topological features of the search landscape. The following theorem shows a relation between the local minima of a meta landscape and the local minima of its underlying landscape.

Theorem 1 *Every local minimum of the meta landscape contains at least one local minimum of the original search landscape and this local minimum is the minimal solution contained in the meta solution.*

Proof Consider a local minimum $\tilde{M}$ from the meta landscape and let $m \in \tilde{M}$ be the solution from $\tilde{M}$ that has the smallest value of f. Hence, $\tilde{f}(\tilde{M}) = f(m)$. Suppose that m is not a local minimum in $L(X, N, f)$. Then, a solution $s \in X$ exists with $f(s) < f(m)$ and $(m, s) \in N$. By the construction $s \notin \tilde{M}$. Hence, there exists a meta solution $\tilde{S} \neq \tilde{M}$ with $s \in \tilde{S}$. From $(m, s) \in N$ follows $(\tilde{M}, \tilde{S}) \in \tilde{N}$, and from $f(s) < f(m)$ we have $\tilde{f}(\tilde{S}) \leq f(s) < f(m) = \tilde{f}(\tilde{M})$. Thus, $\tilde{M}$ is not a local minimum in the meta landscape, which is a contradiction and the result follows.

Note that there is no guarantee that all local minima of the original search landscape are contained in different meta solutions. Thus, a meta landscape can contain significantly fewer local minima than the original search landscape.

The barrier tree is defined with the concept of saddle height [3]. The saddle height between two minima m_1 and m_2 is the minimal cost $h(m_1, m_2)$, such that m_1 and m_2 are still in the same connected component of the subgraph of the search landscape consisting only of the solutions $\{x \in X \mid f(x) \leq h(m_1, m_2)\}$. The following theorem shows how saddle height in the meta landscape is related to saddle height in the original landscape:

Theorem 2 *The saddle height in the meta landscape is a lower bound for the saddle height in the original search landscape.*

Proof Consider two local minima m_1 and m_2 in the search landscape. If there exists a single meta solution $\tilde{M}$ that contains both m_1 and m_2 then the saddle height $\tilde{h}(\tilde{M}, \tilde{M})$ is bounded by $\min\{f(m_1), f(m_2)\}$ and thus, $\tilde{h}(\tilde{M}, \tilde{M}) \leq h(m_1, m_2)$. Now consider the case that there exist two meta solutions $\tilde{M}_1$ and $\tilde{M}_2$ with $m_1 \in \tilde{M}_1$ and $m_2 \in \tilde{M}_2$. Consider a path p between m_1 and m_2 in the original search landscape with $\max_{s \in p}(f(s)) = h(m_1, m_2)$. Path p induces a path $\tilde{p}'$ in the meta landscape by replacing every solution in p with the meta solution that contains it (and by contracting subpaths where all nodes are equal to a same single node). By construction of the meta landscape, the maximal value of f along path $\tilde{p}'$ is less than or equal to $h(m_1, m_2)$. Then, $\tilde{h}(\tilde{M}_1, \tilde{M}_2) \leq \max_{\tilde{s} \in \tilde{p}'}(\tilde{f}(\tilde{s})) \leq \max_{s \in p}(f(s)) = h(m_1, m_2)$.

For investigators of search landscapes, this is important as it allows to draw conclusions about the non-existence of connections below a fixed cost value. The meta landscape might suffer from false positives, i.e., it may show connections when there are none. But there are no false negatives, so that statements about separation in the meta landscape are also valid in the original search landscape.

3.3 Relaxation

In practice, the cost function $\tilde{f}$ cannot be determined exactly without solving the optimization problem. But then an approximation of $\tilde{f}$ might help. We formalize the approximation by allowing for a maximal deviation from the exact values. The relaxed cost function $r\tilde{f} : \tilde{X} \in \tilde{\mathbf{X}} \to \mathbb{R}$ is required to satisfy $\forall \tilde{X} \in \tilde{\mathbf{X}} : \left| \tilde{f}(\tilde{X}) - r\tilde{f}(\tilde{X}) \right| < \epsilon/2$ for some $\epsilon > 0$.

Relaxed cost functions correspond to so-called ϵ-approximate solutions [5], which are available for many optimization problems. The relaxation makes it much harder to draw reliable conclusions from the meta landscape. For the discussion, we introduce the notion of persistence of a minimum as the strength of a perturbation of the cost function that is needed to eliminate the local minimum. This roughly corresponds to the cost difference between the local minimum and a saddle that connects a local minimum with lower costs.

Every minimum of the relaxed meta landscape with a persistence of at least ϵ is still guaranteed to contain a minimum of the original search landscape. Also, the saddle heights can only be interpreted with an additional uncertainty of ϵ. Thus, two solutions m_1 and m_2 are guaranteed to have no connection below a cost value c in the search landscape, if they have no connection below a cost value of $c + \epsilon$ in the meta landscape.

4 Meta Landscapes for Permutation Problems

The use of the meta landscape approach to analyze a search space requires addressing two problem-specific tasks. First, the meta solutions have to be defined in a way that allows to compute the neighborhood relation and the optimal cost value within each meta solution efficiently. Clearly, this depends on the specific optimization problem and the used search operators. Second, the approximation quality of the meta landscape depends on the particular partition of the solutions into meta solutions. In a sense, the quality improves with the granularity of the partition. Thus, the number of meta solutions should be chosen as large as possible while still retaining computational feasibility for both the generation of the meta landscape and its analysis. Further, the partition has to preserve important topological features of the search landscape. Consequently, structures with particularly important features should be placed into different meta solutions. This is a very important property for analysis applications, although it is difficult to guarantee in the individual case and requires profound domain knowledge.

For solving the first task we introduce in the following the permutation tree that allows to easily generate meta solutions for permutation problems. To address the second task, we demonstrate how to control the generation of meta solutions in a way that can be tailored towards specific analysis goals by domain experts, e.g., by applying a heuristic.

4.1 Permutation Trees

For the convenient generation of partitions of the set of permutations of size n, we organize the permutations into a tree structure $PT(n)$ with a height of n (cf. Fig. 1). The tree is called *permutation tree* for further reference. It is a decision tree where at each level one position in the permutation is set. Thus, each leaf of the tree represents one permutation of size n. In practice, this tree is pruned so that in case of symmetries in the cost function of the permutation problem every equivalence class is only present once in the tree.

Every subtree of the permutation tree represents the set of permutations that match in the positions that have been fixed further up in the permutation tree, and differ in the remaining positions. In the following, we characterize each such set of permutations by its fixed, or equivalently by its variable positions. Every set of mutually disjoint subtrees that together include all leaves of the permutation tree, represents a partition of the set of permutations. Thus, we can identify meta solutions with subtrees of the permutation tree.

If we remove a set of k edges from $PT(n)$, the tree is decomposed into $k + 1$ connected components, each of which possibly contains some leaf nodes of $PT(n)$. Components without leaf nodes are discarded. Thus, each such edge set defines a partition of permutations. Likewise, every partition of permutations can be identified with a set of edges that would decompose $PT(n)$ into the corresponding subtrees.

To obtain sets of permutations that are well-suited for use in meta landscapes, we require that for every level i of the permutation tree the variable positions of the

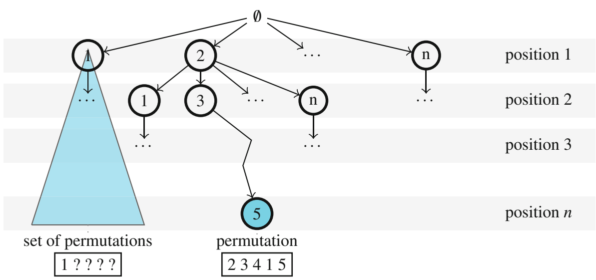

Fig. 1 Permutation tree $PT(n)$. At every level of the tree, one position in the permutations is fixed to the specified element. Every leaf of the tree represents one permutation. A subtree corresponds to a set of permutation that agree at the fixed positions and differ at the remaining ones

permutations form a consecutive range of indices $p_{i+1}, \ldots, p_n$. Given a solution x that belongs to the meta solutions $\tilde{X}$, we consider its set of neighbors $N_x = \{y \in X \mid (x, y) \in N\}$ and investigate the cardinality of $N_x^{in} = N_x \cap \tilde{X}$ and $N_x^{out} = N_x \setminus \tilde{X}$. We want N_x^{in} to be maximal, so that connectivity within a meta solution is as strong as possible. At the same time, we want N_x^{out} to be minimal, so that meta solutions are as separated from each other as possible.

Consider a meta solution $\tilde{X}$ with i fixed positions and $k = (n - i)$ consecutive variable positions. For three common operators on permutations—consecutive swaps, interchange of two positions, 2opt (for details see Schiavinotto et al. [15])—we obtain the following quantities:

Operator	$\lvert N_x^{in} \rvert$	$\lvert N_x^{out} \rvert$
Consecutive swap	$k - 1$	$n - k$
Interchange, 2opt	$(k \cdot (k - 1))/2$	$(n \cdot (n - 1) - k \cdot (k - 1))/2$

These numbers are optimal for sets of permutations with i fixed positions. When changing the partitioning scheme and allowing non-consecutive variable positions in the permutations, we end up with much worse ratios in particular for the consecutive swap and the 2opt operators.

4.2 Heuristic Generation of Partitions by Branching

The generation of a partition by cutting a number of edges in the permutation tree facilitates a branching approach to define and refine such partitions. Initially, we select the out edges of the root node for cutting (cf. Fig. 2). Then, we iteratively select one of these edges and branch it, i.e., we replace it by the out edges of its target node. This is repeated until a partition of sufficient resolution is generated, such that the number of meta solutions stays manageable and at the same time the size of the meta solutions is small enough to provide a meaningful approximation

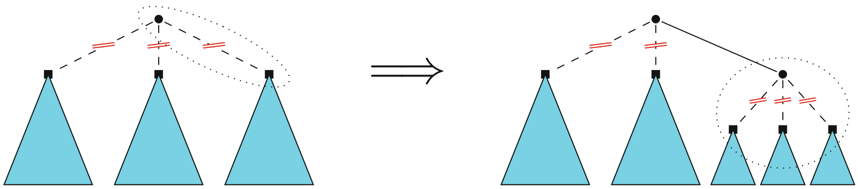

Fig. 2 Branching step in a permutation tree. Initially (left side), some edges (dashed) are selected for cutting. Among these, a branching candidate is determined (circled edge). In the branching step (right side), this candidate is replaced by the outgoing edges of its target

of the original landscape. The process of selecting edges for further refinement can be steered heuristically with respect to the needs of the expert.

For a general analysis of the search landscape, we propose the following branching criteria:

- Branch evenly: Branching is performed so that the resulting meta solutions have nearly equal size.
- Branch by cost: For every meta solution a lower bound for the optimal solution in the meta solution is computed. The meta solution with the lowest bound is branched.
- Branch by cost range: For every meta solution we compute a lower bound and an upper bound. The meta solution with the largest difference between upper and lower bound is branched.

Branching criteria can also be combined, e.g., a smaller size of a meta solution can be weighted against lower costs of a meta solution. However, it seems reasonable to restrict the branching to meta solutions that have a certain minimum size, so that too small meta solutions are avoided.

Furthermore, for an analysis of a landscape it might be interesting to incorporate the global optimum into the branching. In that case the optimum needs to be known or determined beforehand by using an exact solving method. Then, the global optimum can be pre-branched by iteratively branching the meta solution that contains the global optimum until the meta solution falls under a certain size.

5 Results

In this section, we investigate of the proposed meta landscape experimentally and show how it can be applied to TSP and QAP instances. We present the results by using a topological visualization tool for meta landscapes that is a variant of the tool dPSO-Vis [18] with the visual modifications of Bin et al. [1]. dPSO-Vis computes a one-dimensional landscape outline that has the same barrier tree as the meta landscape. We compensate for the differing sizes of the meta solutions by introducing a width for every solution into the visualization (dPSO-Vis considers only individual solutions that naturally are of equal size). As every solution is laid out in an individual horizontal interval, this can be easily achieved by scaling these intervals so their size is proportional to the size of the meta solution. This makes the layout of the topological landscape more robust against changes of the applied branching. It also allows the visual comparison of the visual sizes of different landscape parts—particularly of topological substructures—even if the sizes of the individual meta solutions differ.

5.1 Validation of the Approach

Though we have some theoretical results pertaining to the correctness of the topological approximation with the meta landscape, the quality and amount of detail can only be verified experimentally. For that purpose, we considered small TSP instances with at most 12 cities, so that the complete search landscape was available as a ground truth. The results are demonstrated with the example of a 10-city problem that was generated by randomly placing cities in the plane (cf. Fig. 3). A general drop in the overall landscape height can be observed when using large meta solutions. This is an effect of using the local optimum within a meta solution as the reference cost value for the whole meta solution. However, even when using a meta landscape with only few meta solutions the most persistent topological features are preserved.

5.2 Branching Strategies

We show the results of different branching strategies (cf. Sect. 4.2) for the example of the 52 city TSP instance berlin52 from the TSPLIB [14]. We expect a complex topology, because of the huge magnitude of the search space for this instance. Extrapolating experiences with smaller problem instances, we also expect that no branches in the barrier tree occur outside a small percentage of the best solutions. To estimate the minimal costs within each meta solution, we used the branch-and-bound scheme of Volgenant and Jonker [17].

Figure 4 shows the meta landscapes for the different branching strategies. The meta landscapes differ in the number of branches and the resolution within the interesting parts of the landscape. From the visual size of the meta solutions and

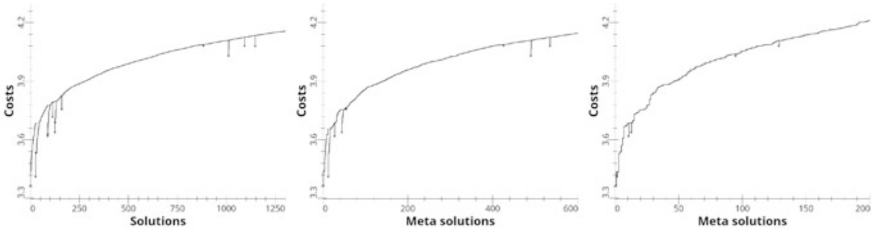

Fig. 3 Interchange operator landscapes of a randomly generated TSP instance with 10 cities (181 440 solutions). Only the lower part of the search landscape is shown as higher parts are topologically not so interesting. The meta landscape were constructed by cutting the permutation tree at the 7-th and 5-th level, respectively, resulting in 30,240 and 1512 meta solutions. (**a**) Ground truth from complete enumeration. (**b**) 7 levels of the permutation tree. (**c**) 5 levels of the permutation tree

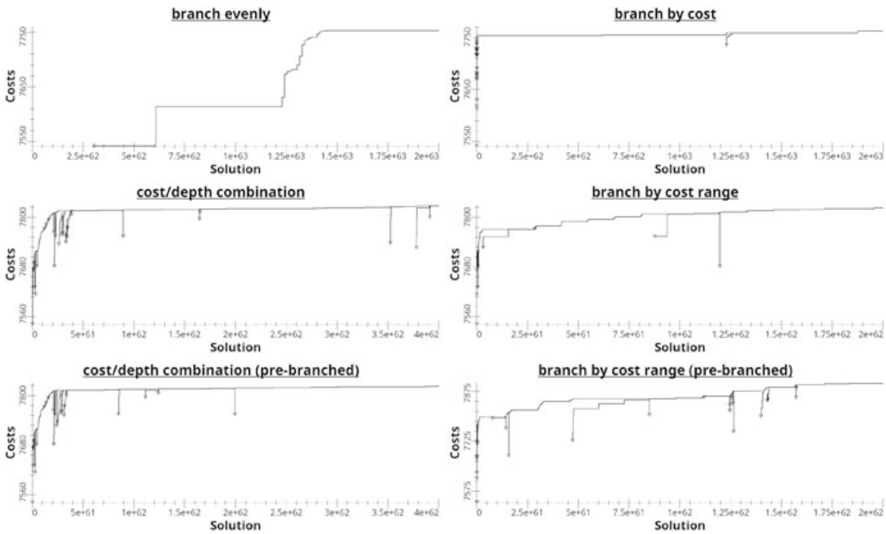

Fig. 4 Swap operator landscapes of the berlin52 instance from the TSPLIB [14] that have been generated with different branching strategies (cf. Sect. 4.2). The complete landscape contains $7.76 \cdot 10^{65}$ solutions. All meta landscapes consist of 50,000 meta solutions with at least 3 and at most 17 fixed permutation positions. Only the topologically relevant part is shown here

their cost values, it can be seen that some branching strategies generate too many meta solutions in parts of the landscape that can be considered as less interesting, i.e., parts with solutions of high costs. This is particularly visible for the strategies branching by cost values and branching evenly. Branching by cost, however, is able to detect a high number of local minima, so that it appears reasonable to include some cost-related metric into the branching. A combined branching strategy that allows level differences between different parts of the permutation tree in relation to the corresponding cost differences (depth/cost combination in Fig. 4) results in the more balanced images and also tends to reveal most branches around the global optimum when compared to other branching strategies. A minimization of the cost intervals for each meta solution appears reasonable from a theoretical point of view (we gain a well-defined persistence per meta solution). However, it does not perform as well as the depth/cost combination. We tested pre-branching the global optimum for the cost/depth combination and the cost range strategy. In both cases, a slightly better focus on the near-optimal part of the landscape can be noticed, but the effect is not very strong.

While the figure shows results for only one example instance, it should be noted that similar differences between the branching strategies have also been found for other TSP instances. Therefore, we suggest to use branching strategies that combine different criteria, e.g., the depth/cost combination, or a branching by cost ranges, and to omit pre-branching.

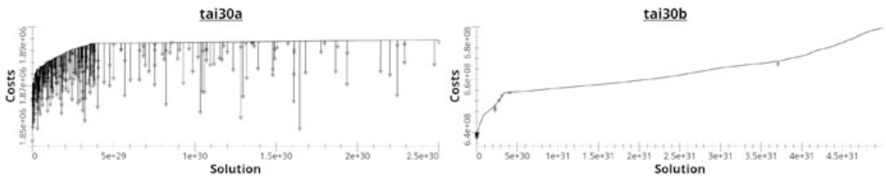

Fig. 5 Landscapes of the problem instances tai30a and tai30b from the QAPLIB [2]. The landscapes have been created with the same parameter set, using branching by a combination of cost and depth with a size of 50,000 meta solutions

5.3 *Comparison of QAP Instances*

Ongoing research tries to categorize QAP instances in order to identify structural properties of them [7], as there can be huge differences between instances. There exist small, but still not optimally solved problem instances, e.g., instance tai30a of Taillard (included in the QAPLIB [2]). However, for the similar instances tai30b which is the sibling instance in the QAPLIB, the global optimum is known. We computed and compared meta landscapes for these two instances. As no well-performing branch-and-bound algorithm for QAP is available, we used probing with multiple local searches to estimate the cost minimum within each meta solution.

Figure 5 contrasts the meta landscapes of both instances. Instance tai30a possesses a much more complicated barrier tree than tai30b. There are many local minima with nearly equal persistence. There also appears a plateau-like structure in the landscape which is indicated by the almost horizontal part of the landscape in the center and the right of the landscape visualization. All this indicates that a bad performance of local search is to be expected. Differently, tai30b contains only few local minima, which also branch away at well-separated cost values. Further, the shape of the fitness landscape might be an indicator for individual structural properties that could be exploited for a well-performing search algorithm.

6 Conclusion

In this paper the concept of meta landscapes was proposed as a well-defined coarse-grained representation of search landscapes. Some properties were shown that allow an interpretation of meta landscapes and to transfer the results back to the original search landscape. Some advice for the computation and usage of meta landscapes was given. In particular, we demonstrated how a heuristically controlled branching scheme could be used to define an application-specific meta landscape.

Since meta landscapes are completely enumerable, topological methods like the barrier tree can be applied to it. Then, topological visualization methods, like dPSO-Vis [18, 19] that build upon the barrier tree, can be used to visually analyse the search landscape. The approach makes permutation problem instances with sizes over 50 accessible for a (visual) analysis of the topological structure of their search landscapes. Thus, the sizes of analyzable search landscapes are increased by several orders of magnitude in comparison to completely enumerable landscapes. This was demonstrated for the example of the TSP and the QAP.

In future work, we want to focus on improved branching strategies for partitioning the search landscape since the branching strategy has a strong influence on the quality of the meta landscape. One possibility would be to pre-sample local minima and to incorporate their distribution within the search landscape into the branching. Also, theoretical results about the quality of the branching are missing. In this paper we focused on permutation problems. However, for subset (or bitstring) problems like MAXSAT, a decision tree can be defined that is similar in structure to the permutation tree. The branching strategies are not dependent on the solutions being permutations, so that the approach should be applicable to subset problems with minimal adaptation. Here, we expect to be able to analyze problems of a size up to 225 variables as these possess similar landscape sizes.

Acknowledgments This work was partly supported by the German Federal Ministry of Education and Research (BMBF) within the project Competence Center of Scalable Data Services and Solutions (ScaDS) Dresden/Leipzig (BMBF grant 01IS14014B).

References

1. Bin, S., Volke, S., Scheuermann, G., Middendorf, M.: Comparing the optimization behaviour of heuristics with topology based visualization. In: Theory and Practice of Natural Computing. Lecture Notes in Computer Science, vol. 8890, pp. 47–58. Springer, Heidelberg (2014)
2. Burkard, R., Karisch, S., Rendl, F.: QAPLIB – a quadratic assignment problem library. J. Glob. Optim. **10**(4), 391–403 (1997)
3. Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Z. Phys. Chem. **216**, 1–19 (2002)
4. Fonlupt, C., Robilliard, D., Preux, P., Talbi, E.G.: Fitness landscapes and performance of meta-heuristics. In: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 257–268. Kluwer Academic Publishers, Berlin (1999)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
6. Hallam, J., Prügel-Bennett, A.: Large barrier trees for studying search. IEEE Trans. Evol. Comput. **9**(4), 385–397 (2005)
7. Herrmann, S., Ochoa, G., Rothlauf, F.: Coarse-grained barrier trees of fitness landscapes. In: International Conference on Parallel Problem Solving from Nature, pp. 901–910. Springer, Heidelberg (2016)
8. Iclanzan, D., Daolio, F., Tomassini, M.: Data-driven local optima network characterization of QAPLIB instances. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 453–460. ACM, New York (2014)
9. McCandlish, D.M.: Visualizing fitness landscapes. Evolution **65**(6), 1544–1558 (2011)
10. Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A study of NK landscapes' basins and local optima networks. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 555–562. ACM, New York (2008)

11. Ochoa, G., Verel, S., Daolio, F., Tomassini, M.: Local optima networks: a new model of combinatorial fitness landscapes. In: Recent Advances in the Theory and Application of Fitness Landscapes, pp. 233–262. Springer, Berlin (2014)
12. Pitzer, E., Affenzeller, M.: A comprehensive survey on fitness landscape analysis. In: Recent Advances in Intelligent Engineering Systems, pp. 161–191. Springer, Heidelberg (2012)
13. Preux, P., Robilliard, D., Fonlupt, C., Karp, R., Steele, J.: Fitness Landscapes of Combinatorial Problems and the Performance of Search Algorithms (1997)
14. Reinelt, G.: TSPLIB—a traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991)
15. Schiavinotto, T., Stützle, T.: A review of metrics on permutations for search landscape analysis. Comput. Oper. Res. **34**(10), 3143–3153 (2007)
16. Stadler, P.F., Schnabl, W.: The landscape of the traveling salesman problem. Phys. Lett. A **161**(4), 337–344 (1992)
17. Volgenant, T., Jonker, R.: A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation. Eur. J. Oper. Res. **9**(1), 83–89 (1982)
18. Volke, S., Middendorf, M., Hlawitschka, M., Kasten, J., Zeckzer, D., Scheuermann, G.: dPSO-Vis: topology-based visualization of discrete particle swarm optimization. Comput. Graph. Forum **32**(3), 351–360 (2013)
19. Volke, S., Bin, S., Zeckzer, D., Middendorf, M., Scheuermann, G.: Visual analysis of discrete particle swarm optimization using fitness landscapes. In: Recent Advances in the Theory and Application of Fitness Landscapes. Emergence, Complexity and Computation, vol. 6, pp. 487–507. Springer, Berlin (2014)
20. Volke, S., Zeckzer, D., Scheuermann, G., Middendorf, M.: A visual method for analysis and comparison of search landscapes. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 497–504. ACM, New York (2015)
21. Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Proceedings of the Sixth International Congress of Genetics, pp. 356–366 (1932)