Fitness Landscape Analysis of NK Landscapes and Vehicle Routing Problems by Expanded Barrier Trees

Bas van Stein, Michael Emmerich, and Zhiwei Yang

LIACS, Leiden University,
Niels Bohrweg 1, 2333-CA Leiden,
The Netherlands
{bvstein,emmerich,zyang}@liacs.nl

Abstract. Combinatorial landscape analysis (CLA) is an essential tool for understanding problem difficulty in combinatorial optimization and to get a more fundamental understanding for the behavior of search heuristics. Within CLA, Barrier trees are an efficient tool to visualize essential topographical features of a landscape. They capture the fitness of local optima and how they are separated by fitness barriers from other local optima. The contribution of this study is two-fold: Firstly, the Barrier tree will be extended by a visualization of the size of fitness basins (valleys below saddle points) using expandable node sizes for saddle points and a graded dual-color scheme will be used to distinguish between penalized infeasible and non-penalized feasible solutions of different fitness. Secondly, fitness landscapes of two important NP hard problems with practical relevance will be studied: These are the NK landscapes and Vehicle Routing Problems (with time window constraints). Here the goal is to use EBT to study the influence of problem parameters on the landscape structure: for NK landscapes the number of interacting genes K and for Vehicle Routing Problems the influence of the number of vehicles, the capacity and time window constraints.

1 Introduction

Much research has been done to combinatorial optimization problems, though we might not even have reached the top of the iceberg yet. Before designing algorithms for a problem, an important task is to understand the properties of the search landscape. Classical calculus has focused mainly on the characterization of continuous search landscapes, whereas landscape analysis tools for discrete search spaces are only discussed more recently also because their application requires fast computing machinery. Our aim is to improve tools for landscape analysis and to study two discrete landscapes with relevance for science and technology, namely the NK landscape problems and the Vehicle Routing problem with time windows (VRPTW). These problems have in common that for a given problem size their difficulty can be scaled by problem parameters, for

M. Emmerich et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Advances in Intelligent Systems and Computing 227,
DOI: 10.1007/978-3-319-01128-8_6,

instance the level of gene interaction K in NK landscapes and the number of vehicles and capacity in VRPTW.

To understand the difficulty of a landscape for local search procedures, the number of local optima is an important characteristic. Beyond this, one may also ask how the attractor basins of local optima are connected with each other. Local Optima Network Analysis [8] and Barrier Tree Analysis [11] provide interesting tools for doing so. In this work we focus on the latter. The idea in Barrier Tree Analysis is to compute a tree that characterizes the topographical structure of a combinatorial landscape. Essentially it provides for each pair of local optima the information on the height of the barrier that separates them from each other. The turning point on an optimal path across this 'fitness barrier' is called a saddle point. For a given landscape this information can be condensed in a hierarchical structure – a barrier tree of the landscape – for which the leaf nodes are local optima and the branching nodes are saddle points.

In Section 2 we provide definition of classical barrier trees and related concepts and refer to some related work. In Section 3 we introduce a new type of barrier trees, we will term *expanded barrier trees* (EBT) and outline an algorithm to compute these. Then, we use the EBT to analyse two problem spaces: Section 4 deals with NK landscapes that are models of evolution of genotypes with interacting genes. Section 5 applies EBT to Vehicle Routing Problems (with Time Windows). Section 6 concludes with a summarizing discussion.

2 Barrier Trees and Related Work

Abstractly, a combinatorial landscape [11] can be defined as a triple $(X, f, \mathcal{N})$, where X denotes a finite, but possibly large, search space, $f : X \to \mathbb{R}$ denotes a fitness function (or height function) that assigns a fitness value to each point in the landscape, and $\mathcal{N} : X \to \wp(X)$ denotes a neighborhood function which declares a neighborhood on X by assigning the set of direct neighbors to points in X. Note, that $\wp(X)$ denotes the set of all subsets (or power set) of X. The neighborhood function is often related to the search heuristic that is used to find an optimal solution. For instance, the set of neighbors can be given by the set of solutions that can be reached from a given point in X by a single mutation or step.

A few formal definitions are required to precisely define barrier trees:

Definition 1 (Path). *Let $N : X \to \wp(X)$ be a neighborhood function. A sequence $p_1, \dots, p_\ell$ for some $\ell \in \mathbb{N}$ and $p_1, \dots, p_\ell \in X$ is called a* path connecting x_1 and x_2, iff $p_1 = x_1$, $p_{i+1} \in N(p_i)$, for $i = 1, \dots, \ell - 1$, and $p_\ell = x_2$. We say that ℓ is the length of the path.

Definition 2. *Let $\mathbb{P}_{x_1,x_2}$ denote the set of all paths between x_1 and x_2.*

Definition 3 (Saddle point). *Let* $\hat{f}(x_1, x_2) = \min_{\mathbf{p} \in \mathbb{P}_{x_1,x_2}}(\max_{x_3 \in \mathbf{p}} f(x_3))$*. A point* s *on some path* $\mathbf{p} \in \mathbb{P}_{x_1,x_2}$ *for which* $f(s) = \hat{f}(x_1, x_2)$ *is called a saddle point between* x_1 *and* x_2*.*

Definition 4 (Basin). *The basin* $B(s)$ *of a point* s *is defined as*

$$B(s) = \{x \in X | \exists \mathbf{p} \in \mathbb{P}_{s,x} : \max_{z \in \mathbf{p}} f(z) \leq f(s)\}.$$

Theorem 1 ([3]). *Suppose for two points* x_1 *and* x_2 *that* $f(x_1) \leq f(x_2)$*. Then, either* $B(x_1) \subseteq B(x_2)$ *or* $B(x_1) \cap B(x_2) = \emptyset$*.*

Theorem 1 implies that the barrier structure of the landscape can be represented as a tree [10] where the saddle points are branching points and the local optima are the leaves.

Example 1. An example of a barrier tree for the search space of a 3-D binary cube is provided in Fig. 1. The search space X is given by the binary numbers of length 3. The hamming neighborhood is applied. Height values are indicated by numbers in the upper part of the nodes.

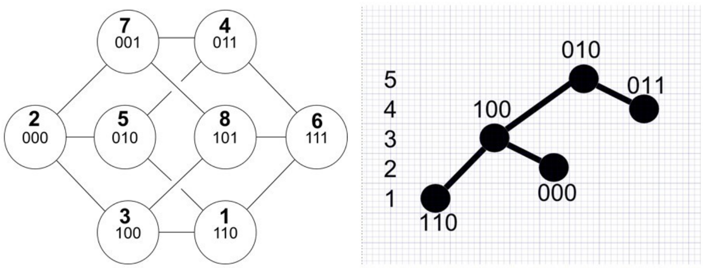
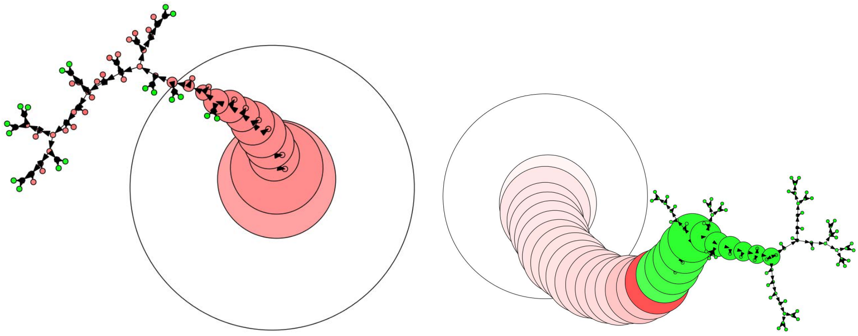


Fig. 1. A binary landscape (left) and its barrier tree (right)

3 Expanded Barrier Trees

The barrier trees describe the notions of saddle points, barriers and basins which give us a vivid landscape visualization of different problems. However, the information of the basin size is not presented yet in barrier trees. Here, we extend the barrier trees to expanded barrier trees, which consist of more details of the landscapes. The addition of basin sizes are attached to each node in the barrier tree. The size of the basin will be represented by the size of the node, which is the logarithm of the number of configurations that belong to the basin. The edge length represents distance in fitness values. After the modification of barrier trees, the expanded barrier trees can be defined as follows:



(a) $c = 3, v = 3$ (b) $c = 3, v = 4$

Fig. 2. Example of Landscapes of VRPTW

Definition 5 (Expanded Barrier Trees). *An expanded barrier tree is a barrier tree with labeled nodes and edges. To each saddle point s in the barrier tree is assigned a natural number* $size(s) = |B(s)|$. *And to each edge is given a value* $length(e) = distance(x_1, x_2)$, *which denotes the distance in height between the saddle points or local optimum* x_1 *and* x_2.

The expanded barrier tree can be graphically represented: Saddle points are represented by means of disks. The radius of the disk shows the size of the basin and the distance of the nodes is expressed as the length of the edge. For better readability of the graphs we also use a coloring of the nodes. Depending on the fitness value we depict the node in a darker or brighter color. The best fitness value (so the global optimum) will be black, while the worst fitness value (the root) will be white. In constrained optimization (e.g. in VRP) a two color scheme is applied: red nodes to describe the infeasible solutions and the green ones indicate that the solution are feasible. Again, the brightness indicates the height of the function value at the node. See Figure 2, for two examples.

In order to generate a barrier tree the *flooding algorithm* (see Algorithm 1) is commonly used [3]. It constructs a barrier tree in discrete landscapes with finite search space $\mathcal{X}$ and linearly ordered search points (e.g. by means of the objective function values). The flooding algorithm builds up the barrier tree in the following way. First, the elements of the search space are sorted in ascending order and send to a queue. Then, the search points are removed one by one from the queue in ascending order and for each point x the following cases are processed:

1. if x_i has no neighbor, it is a local minimum.
2. if x_i has neighbors in only one basin say $B(x_{i_1})$, it also belongs to $B(x_{i_1})$
3. if x_i has neighbors in $n > 1$ basins $B(x_{i_1}, x_{i_2}, ...x_{i_n})$, it is a saddle point. These basins are combined to one $B(x_i)$ and $B(x_{i_1}, x_{i_2}, ...x_{i_n})$ are removed.

After this process, the barrier tree is generated. Note, that if the height function is not injective the flooding algorithm can still be used but the barrier tree may not be uniquely defined. A detailed description of the flooding algorithm is provided with Fig. 1. We indicated with blue color font the parts of the algorithm that have been added to the canonical flooding algorithm, in order to compute expanded barrier trees.

Algorithm 1. Flooding algorithm

Let $x^{(1)}, \ldots, x^{(N)}$ denote the elements of the search space sorted in ascending order.

Let $size(1) \leftarrow 1$, ..., $size(n) \leftarrow 1$.
$i \rightarrow 1; \mathcal{B} = \emptyset$
while $i \leq N$ **do**
  if $N(x_i) \cap \{x^{(1)}, \ldots, x^{(i-1)}\} = \emptyset$ [i. e., $x^{(i)}$ has no neighbor that has been processed.] **then**
    $\{x^{(i)}$ is local minimum$\}$
    Draw $x^{(i)}$ as a new leaf representing basin $B(x^{(i)})$ located at the height of f in the 2-D diagram
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{B(x^{(i)})\}$ {Update set of basins}
  else
    Let $\mathcal{T}(x^{(i)}) = \{B(x^{(i_1)}), \ldots, B(x^{(i_N)})\}$ be the set of basins $B \in \mathcal{B}$ with $N(x^{(i)}) \cap B \neq \emptyset$.
    if $|\mathcal{T}(x^{(i)})| = 1$ **then**
      Let $size(i_1) \leftarrow size(i_1) + 1$.
      $B(x^{(i_1)}) \leftarrow B(x^{(i_1)}) \cup \{x^{(i)}\}$
    else
      $\{x^{(i)}$ is a saddle point$\}$
      Let $size(i) \leftarrow size(i_1) + ... + size(i_N) + 1$.
      Draw $x^{(i)}$ as a new branching point with edges to the nodes $B(x^{(i_1)}), \ldots, B(x^{(i_N)})$. The length of the edges is given by $length((i, i_1)) = f(x^{(i)}) - f(x^{i_1})$, ..., $length((i, i_N)) = f(x^{(i)}) - f(x^{i_N})$, respectively.
      {Update set of basins}
      $B(x^{(i)}) = B(x^{(i_1)}) \cup \cdots \cup B(x^{(i_N)}) \cup \{x^{(i)}\}$
      Remove $B(x^{(i_1)}), \ldots, B(x^{(i_N)})$ from $\mathcal{B}$
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{B(x^{(i)})\}$
    end if
  end if
end while

4 Studies on NK Landscapes

NK Landscapes were introduced by [6] as abstract models for fitness functions based on interacting genes. In NK Landscapes N genes are represented by variables from a finite alphabet, typically of size two. The degree of epistasis (gene interaction) is given by the parameter K. With increasing values of K the ruggedness of an adaptive landscape grows. Besides theoretical biology, NK landscapes have been used as test problem generators for Genetic Algorithms (GAs)[7].

The standard NK Landscapes are fitness functions $F : \{0, 1\}^N \rightarrow \mathbb{R}^+$ that are generated by an stochastic algorithm. Gene interaction data is stored in a randomly generated *epistasis matrix* E, and used to generate a fitness function [1]. The genotype's fitness F is the average of N fitness components F_i, $i = 1, \ldots, N$. Each gene's fitness component F_i is determined by the allele x_i, and also by K alleles at K epistatic genes distinct from i:

$$F(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} F_i(x_i; x_{i_1}, \ldots, x_{i_k}),\ \mathbf{x} \in \{0,1\}^N \tag{1}$$

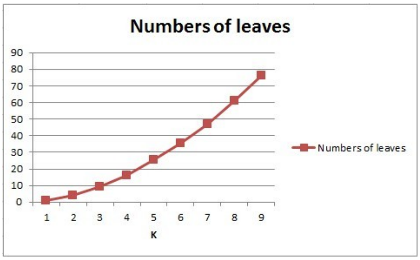
where $\{i_1, \ldots, i_k\} \subset \{1, \ldots, N\} - \{i\}$. There are two ways for choosing K other genes: '**adjacent neighborhoods**', where the K genes nearest to position i on the vector are chosen; and '**random neighborhoods**', where these positions are chosen randomly on the vector. The components $F_i : \{0,1\}^K \rightarrow [0,1)$, $i = 1, \ldots, N$ are computed based on the *fitness matrix* F. For any i and for each of the 2^{K+1} bit combinations a random number is drawn independently from a uniform distribution over $[0,1)$. The algorithm also creates an epistasis matrix E which for each gene i contains references to its K epistatic genes.

Expanded barrier trees of NK landscapes were computed and displayed for $N = 10$ and varying K, in Figure 5 for adjacent neighborhoods and in Figure 6 for random neighborhoods.

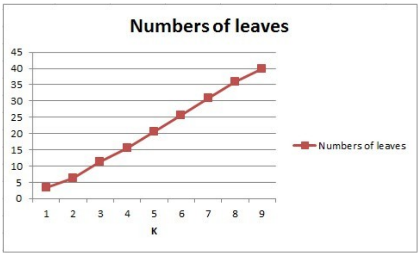
In order to describe our results we will use terminology of axial trees[12,9]. For these trees, that are used as data structures to describe 'natural' trees in geomorphology and biology, branches of different degree are defined. The main branch has degree zero and branches of degree one are side branches of the main branch, and and so on. In the context of EBT we will call 'thick' branches consisting of chains of big nodes to have a higher degree than branches consisting of smaller nodes. However, we will use this definition in a rather informal, descriptive way based on visual appearance. The following observations were found interesting:

1. As expected, the complexity of the NK landscape and thus of the barrier trees grow with K.
2. Landscapes with adjacent neighborhood look slightly more complex than those with random neighborhood.
3. All expanded barrier trees have a relatively large basin for the highest saddle point (white disk). This tendency is getting less, however, as K grows.
4. For each expanded barrier tree, there exists one 'main' branch with small lateral branches which have only one or two local optima. This means a randomized local research cannot be trapped easily, since the basins of these local optima are relatively small and the barriers are not too high.

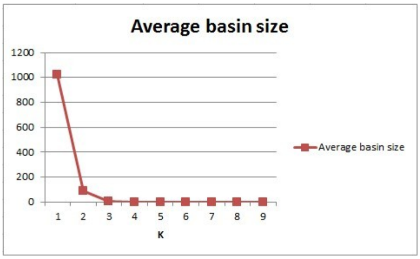
Figure 3 displays the numbers of leaves, average basin size and average edge length of the NK landscape expanded barrier trees with adjacent neighborhoods and random neighborhoods. The numbers of the leaves linearly increase with the number of neighborhoods(K), which means the number of the local optima will grow with K. The average basin size drops dramatically and the average edge length increases exponentially. We can attain that the algorithms are much more



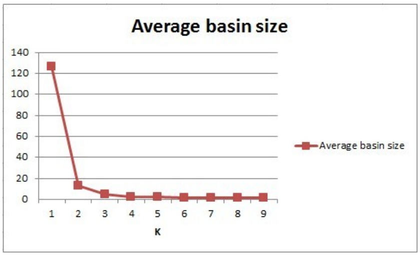
(a) Adjacent Neighborhoods



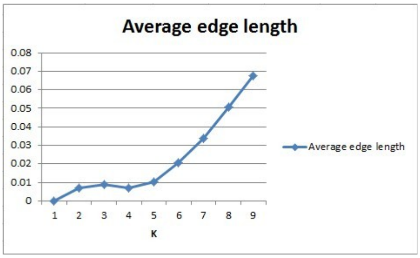
(b) Random Neighborhoods



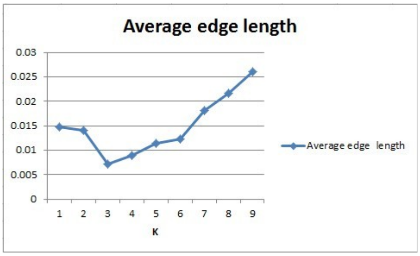
(c) Adjacent Neighborhoods



(d) Random Neighborhoods



(e) Adjacent Neighborhoods



(f) Random Neighborhoods

Fig. 3. The number of leaves , average basin size and average edge length of NK landscape with adjacent neighborhoods and random neighborhoods

easily to trap in the local optima and the energy needed to escape from the local optima increase rapidly. Comparing NK landscape with adjacent neighborhoods and those with random neighborhoods, for the latter the number of leaves of NK landscape with random neighborhoods are lower and the average basin size is smaller and edge lengths are shorter. This means the NK landscape with adjacent neighborhoods present more traps to local search heuristics than those with random neighborhoods.

5 Studies on Vehicle Routing Problems

In this section we will apply expanded barrier tree analysis to a constrained real world optimization problem from logistics. The **Vehicle Routing Problem (VRP)** is a generalization of the Traveling Salesperson Problem (TSP).

The goal is to generate a schedule for a fleet of vehicles that delivers goods from a depot to customers. Each vehicle has a maximum capacity Q and each customer v_i has a demand for a certain amount, q_i, that needs to be delivered. Each VRP problem instance has a special node v_0, called the depot, which is the start and end of each tour. In order to represent solutions with multiple tours as a single sequence, it is assumed that when the depot is visited mid-tour this is equivalent to starting with a new vehicle. The objective in VRP is to minimize the traveling distance and the number of vehicles. In this paper treat the number of vehicles as an equality constraint. The vehicle routing problem is NP hard and several local search heuristics have been proposed for its approximate solution; cf. [4].

In the **Vehicle Routing Problem with Time Windows (VRPTW)** customer nodes v_i have a corresponding time window $[e_i, l_i]$ describing the earliest beginning of service e_i and the latest beginning of service l_i, and a service time s_i. The distance matrix now resembles the traveling time between each two nodes. The depot has a wide time window that starts at $e_0 = 0$ so that all nodes can be serviced before returning to the depot before l_0. Also it has service time $s_0 = 0$. In case a vehicle arrives at node v_i before time e_i it will wait until time e_i. A solution is only valid if for each node v_i, including the depot, the service can be started before time l_i.

To analyze this problem we used a problem instance from the original paper by Dantzig [2] with a sufficient low dimensionality. We did not take one of the *Solomon* instances which are used in many papers, because we need very small problem instances due too the complexity. The problem instance we took has size seven and we can vary the capacity and the number of vehicles. For small capacity there are many infeasible solutions. For every constraint that is not met, we add a penalty to the overall fitness value. The penalty should be high enough to make the best infeasible solution worse than the worst feasible solution.

As indicated before, a configuration of such a *VRP* problem is a string of the customers and several times the central depot. Each time the central depot is in the string means that we use the next vehicle. We define a neighbor of such a configuration, a configuration where two cities are swapped (i.e. two configurations with Cayley distance of 1). Swapping seems a elementary operation to modify traveling salesman like problems such as this VRP problem.

The problem, that was taken, has six customers and a central depot. For simplicity we number the depot and the customers where the depot is defined as number one. Each customer has a distance to each other customer and to the central depot. The distance matrix T is given below:

$$T = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & -1 & 10 & 20 & 25 & 12 & 20 & 2 \\ 2 & 10 & -1 & 25 & 20 & 20 & 10 & 11 \\ 3 & 20 & 25 & -1 & 10 & 25 & 11 & 25 \\ 4 & 25 & 20 & 10 & -1 & 30 & 22 & 10 \\ 5 & 12 & 20 & 25 & 30 & -1 & 30 & 20 \\ 6 & 20 & 10 & 11 & 22 & 30 & -1 & 12 \\ 7 & 2 & 11 & 25 & 10 & 20 & 12 & -1 \end{pmatrix} \quad (2)$$

Extra constraints will now be added to our problem by using Time Windows [5]. We took the instance above and added the following time window constraints: $e_1 = 0$, $l_1 = 999$, $e_2 = 0$, $l_2 = 20$, $e_3 = 15$, $l_3 = 40$, $e_4 = 10$, $l_4 = 30$, $e_5 = 8$, $l_5 = 20$, $e_6 = 20$, $l_6 = 40$, $e_7 = 0$, $l_7 = 5$. These time windows are chosen keeping in mind that there should still exist some feasible solutions.

Four galleries of expanded barrier trees have been computed. Figure 7 shows the results of expanded barrier trees on VRP. And those in Figure 8 shows results of VRPTW. The capacity c and the number of vehicles v are changed from 1 to 4 and from 1 to 5, respectively. Recall that green colors indicate feasible solutions and red colors infeasible ones. Note, that the edge length is not displayed to make the graphics more readable.

The results look very different to the trees of the NK landscape problem. Some interesting findings are:

- As expected, the number of feasible solutions grows with c and v. More surprisingly, for the VRP a single increment of a parameter can cause a transition from a completely feasible search space to a completely infeasible search space or vice versa. This can be observed in the transition from ($c = 2$, $v = 3$) to ($c = 2$, $v = 4$) in Figure 7 or in the transition from ($c = 2$, $v = 2$) to ($c = 3$, $v = 2$).
- This fast transition is not observed the problem with time windows. Rather, in the problems with time windows feasible and infeasible solutions coexist in the search space. An implication of this can be the isolation of feasible components from each other in the landscape. This can be observed in Figure 8 for ($c = 2$, $v = 3$). These isolated local optima can be potential traps for optimization algorithms which do not accept moves to infeasible solutions.
- Again we can characterize the expanded barrier trees as axial trees with a wide zero order branch. Branches of higher order occur typically only in deeper regions of the landscape. This implies that it is relatively unlikely to get trapped in local optima in early stages of a local search. However, as the differences in depth between the small side branches and the main axis are large, it is difficult to escape from these side branches, once they have been entered, even if worsening of the function value would be accepted (as in simulated annealing).

The trees were generated on a Intel Core i72675QM CPU, with two times 2.2 GHz. Only one core was used. The installed working memory is 4.00 GB and the compiler is DEV C++. According to the results, there is no significant difference between the time consumption of VRP trees and VRPTW trees, so we take the VRPTW as an example here. Figure 4 illustrates the time consumption of constructing the expanded barrier trees of VRPTW. From which we can discover the following things:

- As we can see, the time consumption increases rapidly with the number of vehicles (v). And based on this figure, we can easily predict that it will be time-consuming to generating an expanded barrier tree when the number of vehicle grows larger.

– However, what surprises us most is that the time consumption hardly changes when the capacity of the vehicle (c) changes from 1 to 6. This reveals that (c) is not a sensitive parameter we suspected in the process of generating a expanded barrier tree on VRPTW.

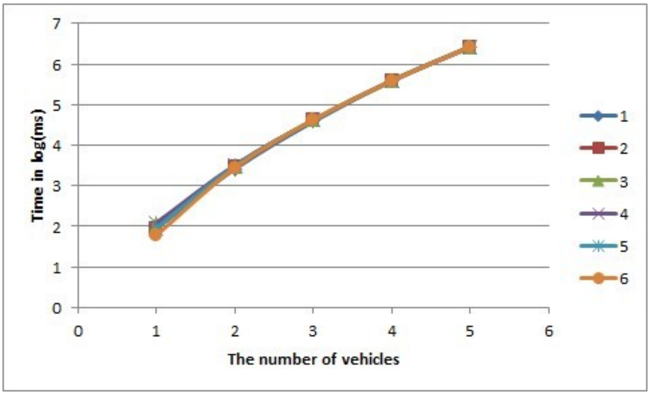


Fig. 4. The time consumption of constructing VRPTW trees

6 Summary and Outlook

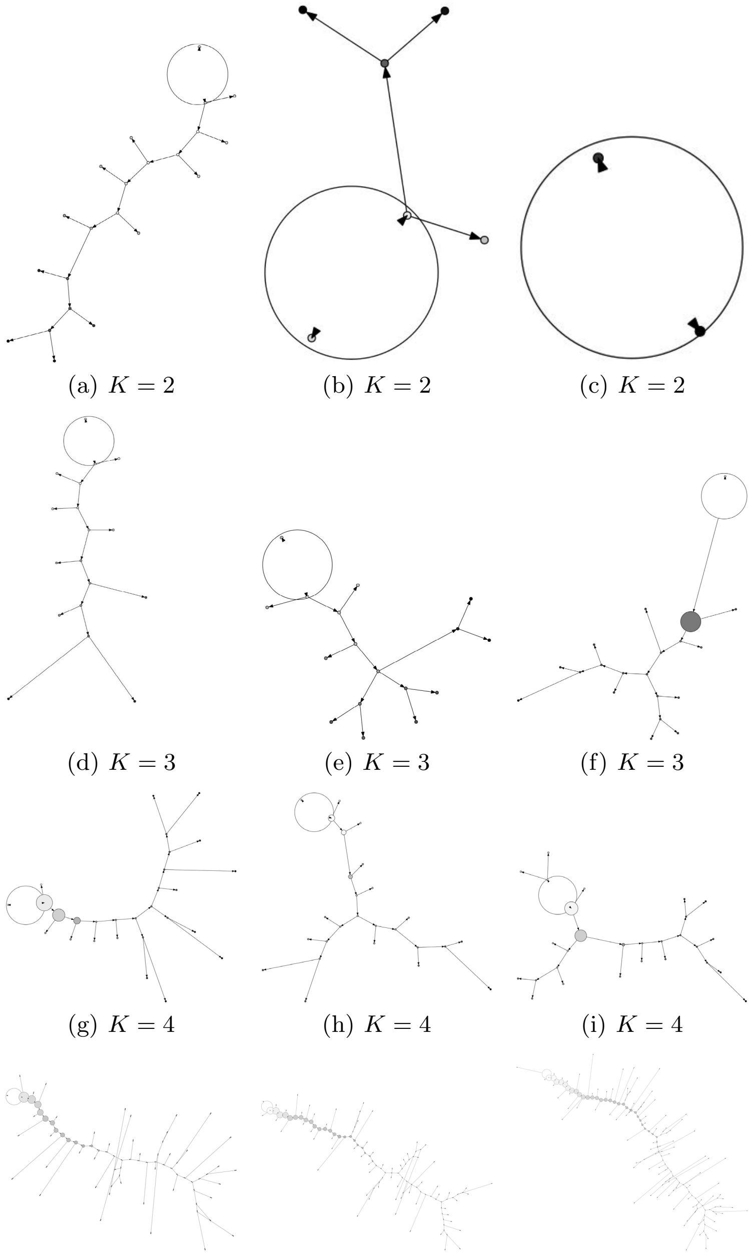
Expanded barrier trees were proposed as a visualization tool for landscape analysis of combinatorial landscapes. As compared to standard barrier trees also the size of the saddle points was defined and a color code for distinguishing infeasible from feasible subspaces.

Expanded barrier trees were computed for two problem classes - NK landscapes and VRP/VRPTW) constraints. For both classes it was shown that the problem difficulty depends crucially on the choice of some parameters.

For NK landscapes the main observation was that for a small value of K there are fewer local optima and branches are only of low order and the highest saddle point has a basin of significant size. For larger values of K the trees get more highly branched and the differences in the size of the saddle points tend to align.

In case of VRP transitions between problems with many feasible solutions and a high fraction of infeasible solutions in the search space were rapid. In case of time window constraints also isolated regions of infeasible solutions occured for parameter values in these critical transitions – a problem that needs to be adressed by optimization algorithm design, e.g. by allowing to tunnel infeasible subspaces or relax constraint penalties. The analysis captured some interesting features of these landscapes, such as disconnected feasible regions.

So far, the studies were confined to small instances, and future work needs to clarify whether or not the insights gained from small models generalize to problems larger input sizes. An interesting observation is also that *expanded*



(a) $K = 2$ (b) $K = 2$ (c) $K = 2$

(d) $K = 3$ (e) $K = 3$ (f) $K = 3$

(g) $K = 4$ (h) $K = 4$ (i) $K = 4$

(j) $K = 7$ (k) $K = 8$ (l) $K = 9$

Fig. 5. Landscapes of NK Adjacent Neighbours

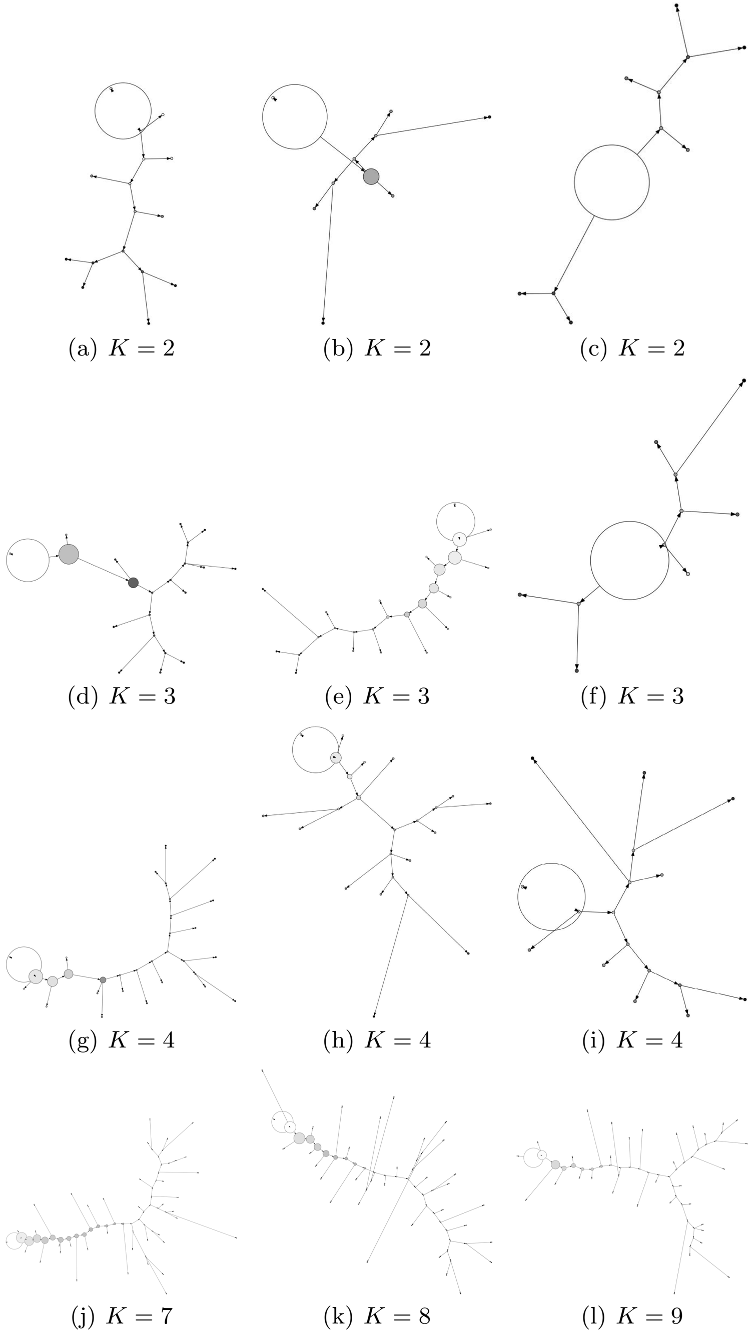


Fig. 6. Landscapes of NK Random Neighbours

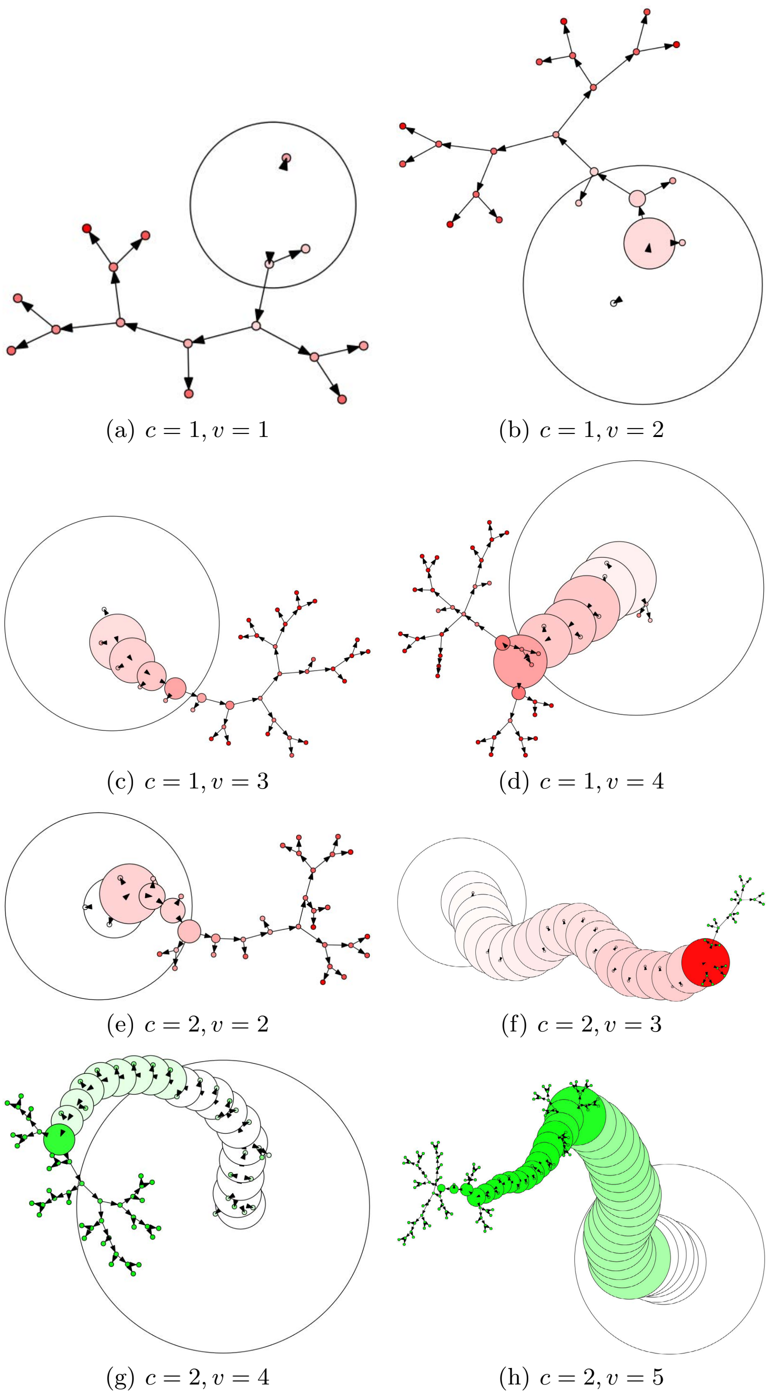
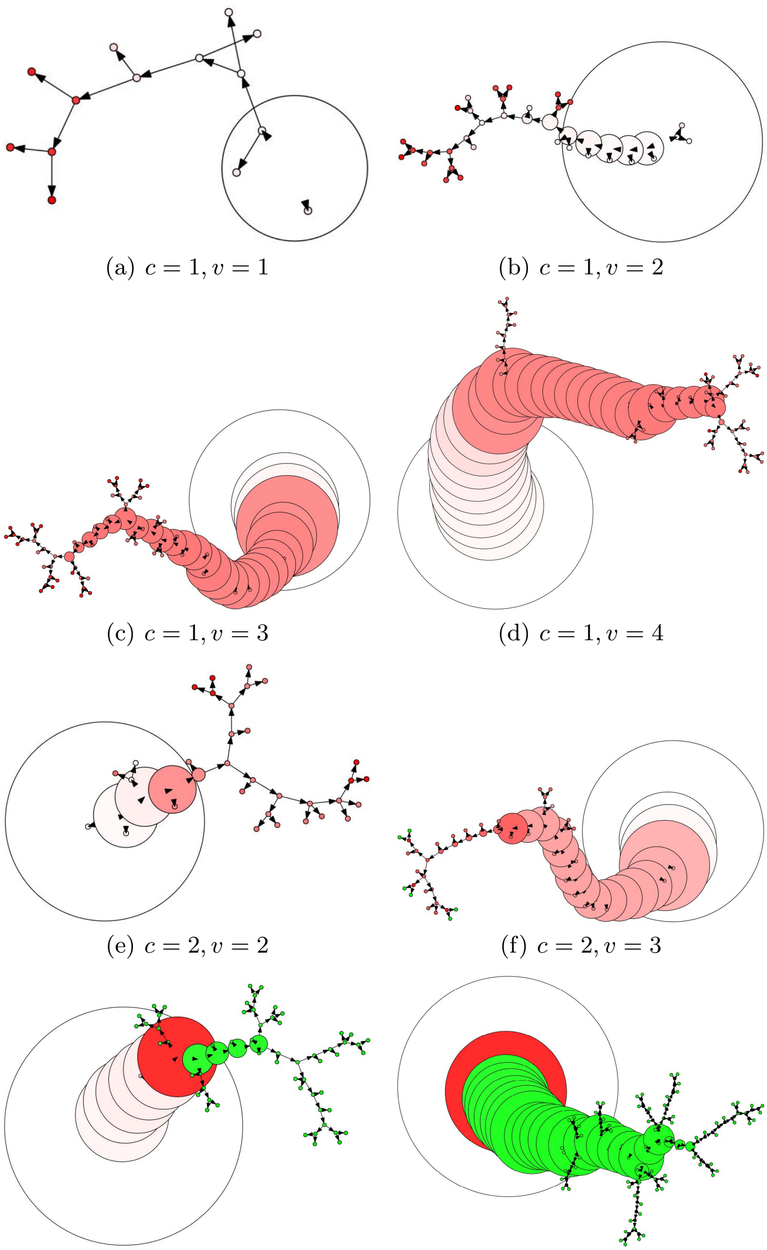


Fig. 7. Landscapes of VRP



(a) $c = 1, v = 1$ (b) $c = 1, v = 2$

(c) $c = 1, v = 3$ (d) $c = 1, v = 4$

(e) $c = 2, v = 2$ (f) $c = 2, v = 3$

(g) $c = 2, v = 4$ (h) $c = 2, v = 5$

Fig. 8. Landscapes of VRPTW

barrier trees can be characterized in the terminology of axial trees, an aspect that could be elaborated further. Moreover, alternative visualization techniques such as local optima network should be studied for the same landscapes (cf. [13]) in future work.

Acknowledgements. The authors acknowledge support of Agentschap NL, The Netherlands within the DELIVER project for their kind support. The software used for generating the tree visualizations is based on the program GraphViz. The source code will be made available under `http://natcomp.liacs.nl`.

References

1. Altenberg, L.: Nk-fitness landscapes. In: Bäck, T., et al. (eds.) Handbook of Evolutionary Computation. Oxford University Press (1997)
2. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management Science 6(1), 80–91 (1959)
3. Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Zeitschrift für Physikalische Chemie 216, 155 (2002)
4. Gambardella, L.M., Taillard, É., Agazzi, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: New Ideas in Optimization, ch. 5, pp. 63–76. McGraw-Hill (1999)
5. Kallehauge, B., Larsen, J., Madsen, O.B.G., Solomon, M.M.: Vehicle routing problem with time windows. Column Generation, 67–98 (2005)
6. Kauffman, S.A.: The origins of order: Self-organization and selection in evolution. Oxford University Press, USA (1993)
7. Li, R., Emmerich, M.T.M., Eggermont, J., Bovenkamp, E.G.P., Bäck, T., Dijkstra, J., Reiber, J.H.C.: Mixed-integer nk landscapes. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 42–51. Springer, Heidelberg (2006)
8. 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 (2008)
9. Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Springer (1996)
10. Rammal, R., Toulouse, G., Virasoro, M.A.: Ultrametricity for physicists. Reviews of Modern Physics 58(3), 765 (1986)
11. Stadler, P.: Towards a theory of landscapes. In: Complex Systems and Binary Networks. Lecture Notes in Physics, vol. 461, pp. 78–163. Springer, Heidelberg (1995)
12. Strahler, A.N.: Quantitative analysis of watershed geomorphology. Transactions of the American Geophysical Union 38(6), 913–920 (1957)
13. Tomassini, M., Daolio, F.: A complex-networks view of hard combinatorial search spaces. In: Tantar, E., Tantar, A.-A., Bouvry, P., Del Moral, P., Legrand, P., Coello Coello, C.A., Schütze, O. (eds.) EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation. SCI, vol. 447, pp. 221–244. Springer, Heidelberg (2013)