Deconstructing the Big Valley
Search Space Hypothesis

Gabriela Ochoa and Nadarajen Veerapen

Computing Science and Mathematics, University of Stirling, Scotland, UK.

Abstract. The big valley hypothesis suggests that, in combinatorial optimisation, local optima of good quality are clustered and surround the global optimum. We show here that the idea of a single valley does not always hold. Instead the big valley seems to de-construct into several valleys, also called 'funnels' in theoretical chemistry. We use the local optima networks model and propose an effective procedure for extracting the network data. We conduct a detailed study on four selected TSP instances of moderate size and observe that the big valley decomposes into a number of sub-valleys of different sizes and fitness distributions. Sometimes the global optimum is located in the largest valley, which suggests an easy to search landscape, but this is not generally the case. The global optimum might be located in a small valley, which offers a clear and visual explanation of the increased search difficulty in these cases. Our study opens up new possibilities for analysing and visualising combinatorial landscapes as complex networks.

1 Introduction

In the mid 1990s, it was conjectured that the search space of travelling salesman instances had a "globally convex" or "big valley" structure, in which local optima are clustered around one central global optimum [3]. This globally convex structure has subsequently been observed in other combinatorial problems such as the NK family of landscapes [7, 8], graph bipartitioning [13], and flowshop scheduling [21]. Under this view, there are many local optima but they are easy to escape from, with the coarse level gradient leading to the global optimum (see Fig. 1). This hypothesis has become generally accepted and has inspired the design of modern search heuristics.

We argue that this view of combinatorial search spaces is not complete. We challenge the existence of a single valley, and present compelling and visual evidence of examples where the big valley de-constructs into several valleys, also called 'funnels' in the study of energy surfaces in theoretical chemistry [14, 9]. The multi-funnel concept implies that local optima are organised into clusters, so that a particular local optimum largely belongs to a particular cluster.

Using the travelling salesman problem (TSP) as a case study, we found that this decomposition into clusters does not only occur near the global optimum as has been observed recently [6, 19]. It occurs earlier on in the search process, even among local optima tours with relatively high costs. This finding improves our
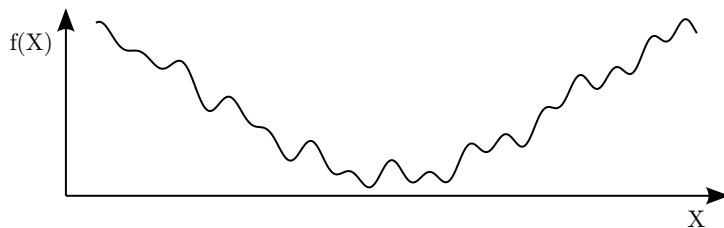
Fig. 1: Depiction of the 'big-valley' structure.

understanding of search difficulty in combinatorial optimisation. It explains why, when using current local search heuristics, random restarts are generally required to consistently find globally optimal solutions. When trapped in a sub-optimal funnel, a local search heuristic will not be able to escape even with relatively large random perturbations. This insight will foster research into more informed escaping and *tunnelling* mechanisms [24, 6, 17].

We use the local optima networks model to analyse and visualise the big valley deconstruction. Local optima networks compress the whole search space into a graph, where nodes are local optima and edges are transitions among them with a given search operator [18, 20, 23]. Local optima are key features of fitness landscapes as they can be seen as obstacles for reaching high quality solutions. The local optima networks model emphasises the number, distribution and most importantly, the connectivity pattern of local optima in the underlying search space. They are therefore an ideal tool for modelling and visualising the big valley structure.

We propose a new and effective sampling procedure for extracting the network data of large instances. Local optima and escape edges are collected from several runs of *Chained Lin-Kerninghan*, a state-of the-art TSP heuristic [12]. This data is gathered to construct the local optima networks.

The remainder of this article is organised as follows. The next section gives an overview of Chained Lin-Kerninghan. Section 3 defines the local optima network model considered, and describes the procedure for extracting the data and constructing the networks. Section 4 describes the TSP instances studied. Section 5 presents the analysis and visualisation of the extracted local optima networks. Finally, Section 6 summarises our main findings and suggests directions of future work.

2   Chained Lin-Kernighan

Lin-Kernighan (LK) [10], is a powerful and well-known heuristic for solving the TSP. For about two decades, it was the best local search method, and nowadays it is a key component of state-of-the-art TSP solvers. LK search is based on the idea of k-changes: take the current tour and remove k different links from it, which

are then reconnected in a new way to achieve a legal tour. A tour is considered to be 'k-opt' if no k-change exists which decreases its length. Fig. 2a illustrates a 2-change move. LK applies 2, 3 and higher-order k-changes. The order of a change is not predetermined, rather k is increased until a stopping criterion is met. Thus many kinds of k-changes and all 3-changes are included. There are many ways to choose the stopping criteria and the best implementations are rather involved. Here, we use the implementation available in the Concorde software package [1], which uses *do not look* bits and candidate lists.
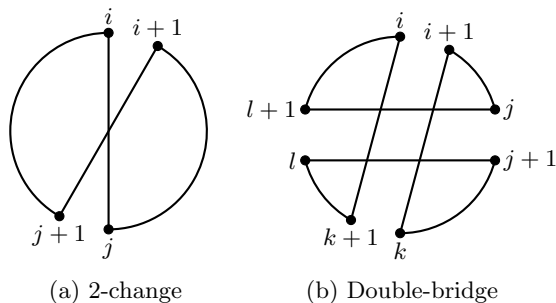

(a) 2-change   (b) Double-bridge

Fig. 2: Illustration of tours obtained after 2-change and double-bridge moves.

The overall tour-finding strategy using LK-search was to repeatedly start the basic LK routine from different starting points keeping the best solution found. This practice ended in the 1990s with the seminal work of Martin, Otto and Felten [12], who proposed the alternative of *kicking* (perturbing slightly) the LK tour and reapplying the algorithm. If a better tour is produced, we discard the old LK tour and keep the new one. Otherwise, we continue with the old tour and kick it gain. This simple yet powerful strategy is nowadays best known as *iterated local search* [11]. It was named *Chained Lin-Kernighan* (Chained LK) by Applegate et al. [2], who also provided an improved implementation to solve large TSP instances.

The kick or escape operator in Chained-LK is a type of 4-change, named *double-bridge* by Martin et al. [12] (drawn in Fig. 2b). It consists of two improper 2-changes, each of which is a 'bridge' as it takes a legal, connected tour into two disconnected parts. The combination of both bridges, must then be chosen in order to produce a legal final tour.

3 Local optima networks for TSP

To construct the networks we need to define their nodes and edges. The definition is closely linked to the methodology for extracting the network data, which is based on a number of runs of the Chained-LK algorithm described above.

Clearly, a full enumeration of the local optima for TSP instances of non-trivial size becomes unmanageable. Therefore, the networks are based on a sample of high-quality local optima in the search space. We first provide some basic definitions, below, before describing the sampling algorithm.

3.1 Definitions

Definition 1. A tour is a *local optimum* if no tour in its neighbourhood is shorter than it. The neighbourhood is imposed by LK-search, which considers variable values of k. The local optimality criterion is, therefore, rather stringent. Only a small proportion of all possible tours are LK-optimum. The set of local optima is denoted by LO.

Definition 2. Edges are directed and based on the double-bridge operator. There is an *escape* edge from local optimum LO_i to local optimum LO_j, if LO_j can be obtained after applying a double-bridge kick to LO_i followed by LK-Search. The set of escape edges is denoted by E_{esc}.

Definition 3. The *local optima network, LON*, is the graph $LON = (LO, E_{esc})$ where nodes are the local optima LO, and edges E_{esc} are the escape edges.

3.2 Gathering network data

To extract the network data, we instrumented the Chained-LK implementation of Concorde (see Algorithm 1). We simply store, in LO, every unique local optima obtained after an LK application, and create and store, in E_{esc}, an edge between the starting and end optima after a double-bridge move.

Data: I, a TSP instance
Result: LO, the set of local optima,
$\qquad\quad E_{esc}$, the set of edges between local optima
$n \leftarrow \text{numberOfCities}(I);\ LO \leftarrow \{\};\ E_{esc} \leftarrow \{\}$
for $i \leftarrow 1$ **to** 100 **do**
$\quad$ $s \leftarrow \text{initialSolution}()$
$\quad$ $s \leftarrow \text{LK}(s)$
$\quad$ $LO \leftarrow LO \cup \{s\}$
$\quad$ **for** $k \leftarrow 1$ **to** n **do**
$\quad\quad$ $s_{start} \leftarrow s$
$\quad\quad$ $s_{end} \leftarrow \text{applyKick}(s)$
$\quad\quad$ $s_{end} \leftarrow \text{LK}(s_{end})$
$\quad\quad$ $LO \leftarrow LO \cup \{s_{end}\}$
$\quad\quad$ $E_{esc} \leftarrow E_{esc} \cup \{(s_{start}, s_{end})\}$
$\quad\quad$ **if** $\text{fitness}(s_{end}) < \text{fitness}(s_{start})$ **then** $s \leftarrow s_{end}$
$\quad$ **end**
end
Algorithm 1: Local optima network sampling for 100 runs of Chained-LK.

A hundred independent runs of Chained-LK are executed. We chose to use two different starting mechanisms, one producing "better" solutions, the other "worse" solutions, to have a broader picture of the search space. Half of the runs start from a relatively good solution, built using the Quick-Borůvka method. The latter is the default initialisation for Concorde's Chained-LK and is based on the minimum-weight spanning tree algorithm of Borůvka [15]. The other half starts from a random solution.

Each run performs n kicks, where n is the size of the tour (number of cities). The default kicking procedure in Concorde is used: the edges involved in the double bridge are selected using random walks along connected vertices.

Since nodes and edges are collected from a combination of several runs, it is possible that each of them is found more than once. Therefore, weights could be associated to edges indicating the number of times they were encountered. We recorded such weights, but chose to analyse unweighted networks. Future work will consider this information in the analysis.

4 Selected TSP instances

Our study considers four TSPLIB [22] instances of a few hundred cities belonging to different types (see Table 1). By exploring and comparing the local optima networks of instances of similar size, we aim to discover structural differences distinguishing the hard from the easy to solve instances.

Table 1: Selected TSP instances

Property	Instances			
	att532	u574	rat575	gr666
Cities	532	574	575	666
Edge Weight Type	ATT	EUC-2D	EUC-2D	GEO
Description	US cities	Drilling problem	Rattled grid	World cities
Optimum	27686	36905	6773	294358
Concorde run time (s)	8.9	3.8	18.9	6.5
Concorde B&B nodes	5.2	1.7	17.7	3.2
Chained-LK success rate	0.06	0.47	0.01	0.04

The types of edge weights are as follows. EUC-2D refers to the Euclidean distance of points in a 2D plane rounded to the nearest integer. ATT refers to a pseudo-Euclidean distance: the sum of the squares is divided by 10 and the square root of this value is then rounded to an integer. GEO refers to the integer

geographical distance computed from latitude and longitude coordinates on the surface of a sphere representing an idealized Earth.

The bottom portion of Table 1 gives information on the solving difficulty of each instance. Specifically, we report the mean run time and the mean number of branch-and-bound nodes required by Concorde (interfaced with IBM ILOG CPLEX 12.6) to solve the instances to optimality on a 3.4 GHz Intel Core i7-3770 CPU across 10 runs. Although Concorde is an exact solver, the means are computed since randomised heuristics, including Chained-LK, are used. This leads to different execution times and branch-and-bound trees. The last row reports the success rate of the 100 Chained-LK runs used for extracting the network data (described in Section 3.2). By success rate, we mean the ratio of runs that found at least one global optimum. According to this information, the easiest instance to solve is u574 (by far) and the hardest is rat575.

5   Results

When extracting local optima networks from large instances, it is important to decide which sample to consider. We chose here to analyse two sets: (i) the whole set of local optima collected with the procedure described in Section 3.2, and (ii) the subset containing the best 10 % local optima according to fitness. For each TSP instance in Table 1, we consider the two sets and construct the local optima networks as defined in Section 3.

Results are presented in the following two subsections, which conduct a network analysis and a fitness distance correlation analysis, respectively.

5.1   Network analysis and visualisation

Over the years, an extensive set of tools – mathematical, computational, and statistical – have been developed for analysing and understanding networks [16]. We select here a set of network features (see Table 2) which we consider relevant to search dynamics.

Table 2: Main local optima network features.

Feature	Description
nv	Total umber of vertices (local optima)
ne	Total number of edges
n_{go}	Number of different global optima
nc	Number of connected components (or clusters)
c_{go}	Cluster containing the global optimum (or optima), where the clusters are ordered by decreasing size.

We argue that the decomposition into clusters (connected components) is one of the most relevant features impacting search. Indeed, we hypothesise that

Table 3: Network metrics (as described in Table 2) for the four TSP instances and the two local optima samples: *all* and *best 10 %*.

	att532		u574		rat575		gr666	
	all	*best 10 %*	*all*	*best 10 %*	*all*	*best 10 %*	*all*	*best 10 %*
nv	35,512	3,678	37,780	3,842	41,536	4,805	46,021	4,611
ne	37,730	4,435	40,161	4,660	44,643	5,842	47,939	5,039
n_{go}	2	2	4	4	2	2	2	2
nc	6	7	8	5	69	47	53	25
c_{go}	1	2	1	1	60	8	4	2

the notion of multiple funnels, originally studied in theoretical chemistry [14, 9], and more recently also in combinatorial optimisation [6, 17], is captured by the connected components in the networks studied. Specifically, funnels correspond to connected components. Once trapped in a connected component, it is not easy for the search process to hop to another component. There are no connections among components with the underlying escaping mechanism. We, therefore, explore in detail the connected components decomposition of the studied networks.

Table 3 reports the main network features for each instance and local optima sample. All instances have more than one global optima, and all decompose into several clusters. Indeed, several components are found on both samples, indicating that the deconstruction occurs not only among solutions near the global optima, but early on in the search process (solutions with higher costs). The last row in the table (c_{go}) shows that global optima are not always found in a large connected component.

It is interesting to note that for the hardest instance studied, rat575 (see the bottom of Table 1 for an indication of search difficulty), the global optima were not found in any of the 5 largest connected components. They are located in cluster number 60 when considering the whole sample and cluster 8, when considering the best 10 % local optima. On the other hand, for the easiest instance, u574, the global optima are found in the largest connected component for both samples. Table 4 reports sizes (as percentages) of the 5 largest connected components for each instance and local optima sample. Bold fonts indicate the component containing the global optima. As mentioned before, global optima are not found in the top 5 connected components of instance rat575, they are located in the 8th component, which contains only 2.67 % of the local optima sample.

A useful approach to explore the structure of networks is to visualise them. Software for analysing and visualising networks is currently available in various languages and environments. Here we use the R statistical language together with the igraph package [4]. The graph layout algorithm used is the Fruchterman and Reingold method [5], which is based on exploiting analogies between the relational structure in graphs and the forces among elements in physical systems. The heuristic is concerned with drawing graphs according to some generally

Table 4: Sizes (as percentages) of the top 5 connected components for the four TSP instances and the two local optima samples: *all* and *best* 10%. Bold fonts highlight the connected component containing the global optima. For instance rat575, the global optima are located in the 8th component, which contains only 2.67% of the local optima sample

	att532		u574		rat575		gr666	
	all	*best 10%*	*all*	*best 10%*	*all*	*best 10%*	*all*	*best 10%*
c_1	**93.62**	50.33	**58.56**	**85.84**	17.55	33.47	17.20	29.75
c_2	2.17	**48.15**	15.97	11.63	4.93	7.62	7.12	**13.42**
c_3	1.12	1.20	10.30	1.54	4.89	3.79	7.05	12.67
c_4	1.04	0.19	8.72	0.60	2.03	3.33	**6.22**	6.66
c_5	1.04	0.08	3.20	0.39	2.01	2.93	3.90	4.49

accepted aesthetic criteria such as a) distribute the vertices evenly in the frame (a circle in this case), b) minimise edge crossings, c) make edge lengths uniform, and d) reflect inherent symmetry [5].

In order to have manageable images, we plotted the networks corresponding to the subset containing the best 10% local optima. We also pruned some of the nodes of degree one, and removed self-loops for improved visibility. Figure 3 shows the local optima for instances att532 and u574, and Figure 4 shows the networks for rat575 and gr666. Nodes are LK-search local optima and edges represent escape transitions according to double-bridge moves.

We decorated the network images according to the two most relevant features impacting search dynamics: fitness and connectivity. The fitness of a solution is reflected by its node size, with size inversely proportional to tour cost (so the best solutions are larger in size). The connected components are distinguished with different colours: red shows the largest connected component, blue the 2nd largest and so on, as indicated in the legends of Figures 3 and 4. Global optima nodes are highlighted with a yellow outline.

The networks show strikingly different structures. In instance att532 (Fig. 3, top), the two largest connected components (red and blue) show similar sizes, with the remaining components having small sizes (see also Table 4 for percentages). The two global optima are located in the blue component. In instance u574 (Fig. 3, top), the largest component (red) clearly dominates, containing the four global optima and many good local optima as indicated by the node sizes. This is consistent with the fact that u574 is the easiest instance to solve, as indicated in the bottom of Table 1.

In the first two instances considered (Fig. 3), the two largest components (red and blue) dominate the network images. This is not the case for instances rat575 and gr666 (Fig. 4), where the smaller connected components are more visible. In rat575, the global optima are not found in the top 5 components. Instead, the two global optima are located in component number 8, visualised in dark green
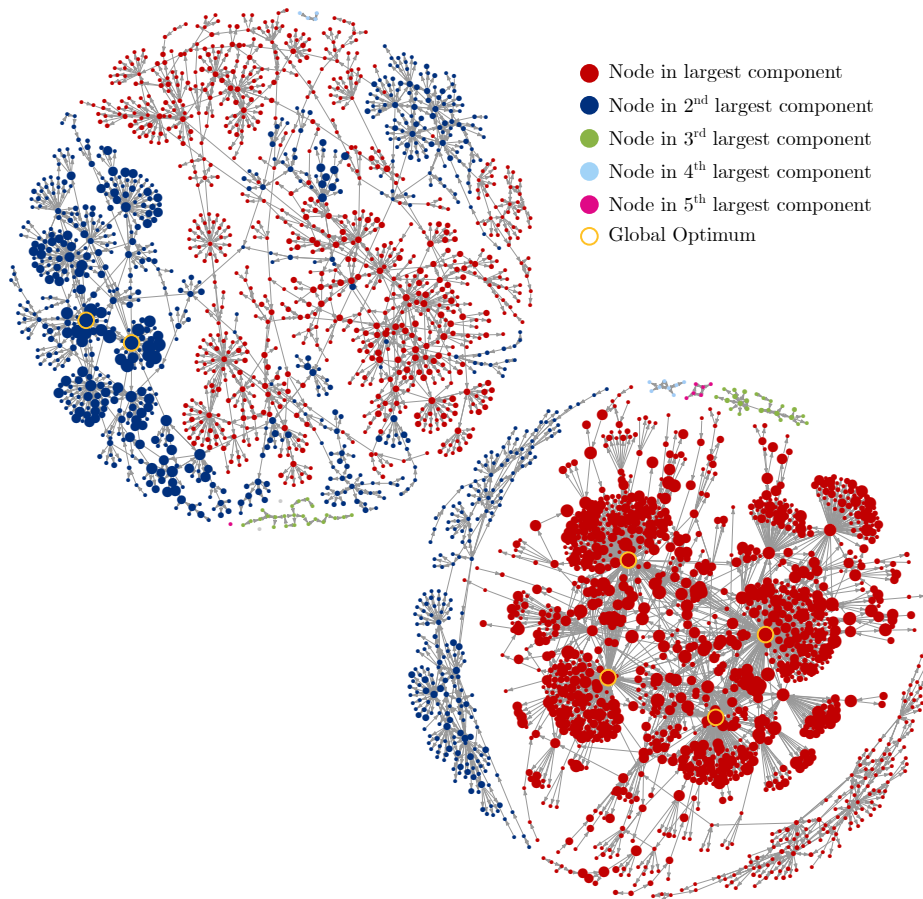
Fig. 3: Local optima networks for att532 (top) and u574 (bottom). Nodes are LK-search local optima, and edges represent escape transitions according to double-bridge moves. Node colours identify connected components as indicated in the legend, while node sizes are inversely proportional to tour cost (so the best solutions are larger in size). Global optima nodes are highlighted with a yellow outline.
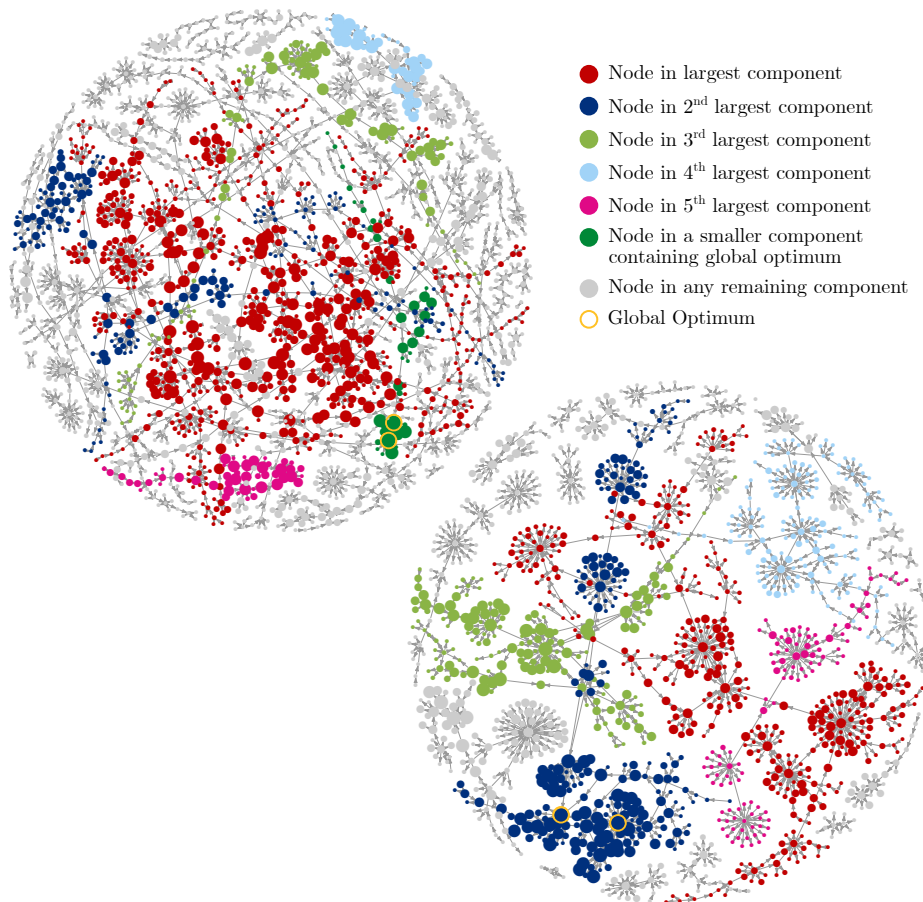
Fig. 4: Local optima networks for rat575 (top) and gr666 (bottom). Nodes are LK-search local optima, and edges represent escape transitions according to double-bridge moves. Node colours identify connected components as indicated in the legend, while node sizes are inversely proportional to tour cost. Global optima nodes are highlighted with a yellow outline.

at the bottom right of the network plot. This is a small component containing only 2.67% of the local optima sample: this provides a clear visual indication of the increased search difficulty of this instance. For the gr666 instance, the global optima are located in the 2nd largest connected component (blue) visualised at the bottom left portion of the image. This component contains 13.42% of the local optima, suggesting an easier to search instance despite having a larger number of cities.

This study only considers instances where the number of connected components was less than the number of runs. Yet, the maximum number of components that could be discovered by the sampling method corresponds to the number of runs if no local optimum is repeated in any two runs. It is nevertheless possible that some instances actually have many more components than this number. It is also important to note that the sampling mechanism, including the parameter values for the number of kicks and runs, introduces a bias generating an approximation of the search space and not the complete picture.

5.2 Fitness-distance analysis

While the network analyses provide insight into the connected nature of the search space, it is also useful to examine the landscapes through more traditional tools. In particular, we now look at the relationship between fitness and bond distance [3]. The latter is defined as the difference in the number of common edges, or bonds, between two tours. It is computed by subtracting the number of common edges from the number of cities. We specifically consider the distance between a single randomly chosen global optimum and the other local optima. Let us note that the global optima for each instance share the overwhelming majority of their edges. The bond distances between the global optima are 2 for att532, $\{2, 4, 6\}$ for u574, 3 for rat575 and 13 for gr666. It is thus logical for them to appear within the same component.

Figure 5 presents the fitness-distance plots for the *best 10 %* sub-sampling that is represented in the local optima networks and reuses the same colour-component correspondence. Each of the 5 largest components is plotted in a separate facet. When the global optima are found in a smaller component, the points of the latter have their own facet. Any remaining components are grouped in one final facet. Each plot also displays all the local optima across components in the background.

Figure 6 provides a similar view of the local optima but considers (almost) complete samples. Points with fitness above the 95th percentile are not plotted because they are very spread out and thus interfere with visualisation. Components, however, are computed with respect to all the points in the *all* sample. Points in common with those in Figure 5 are highlighted with the same colour scheme, with the aim of exploring the correspondence of clusters between the two studied samples.
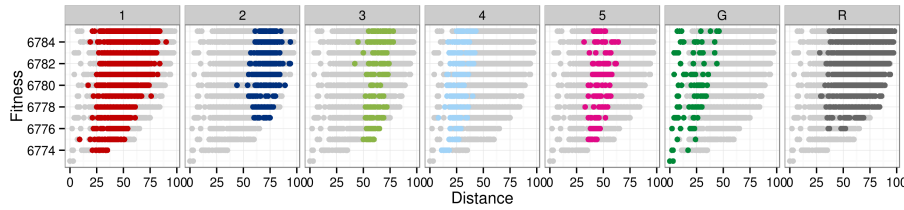
In the *best 10 %* sampling, smaller components containing a few solutions are artefacts of the arbitrary threshold and actually form part of larger distinct clusters. For att532, even though this is not visible due to points overlapping, all the
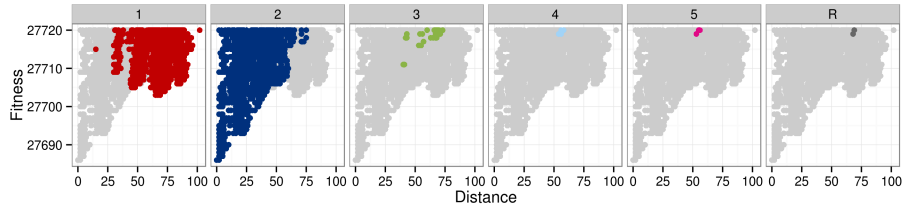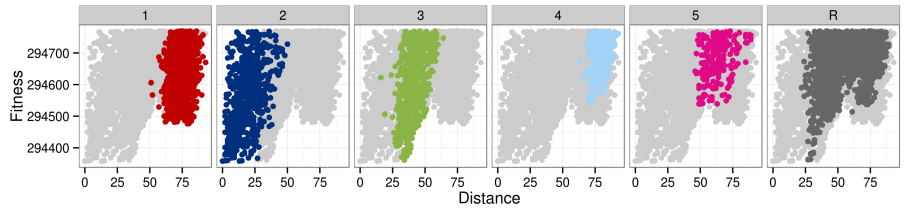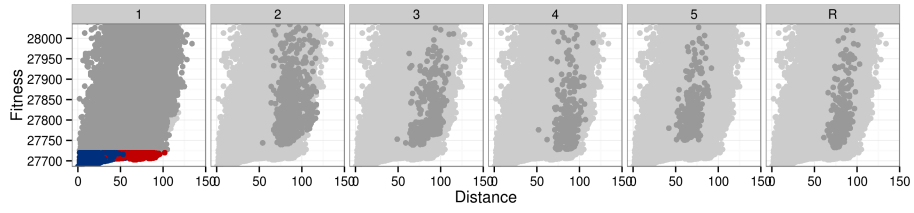
(a) att532
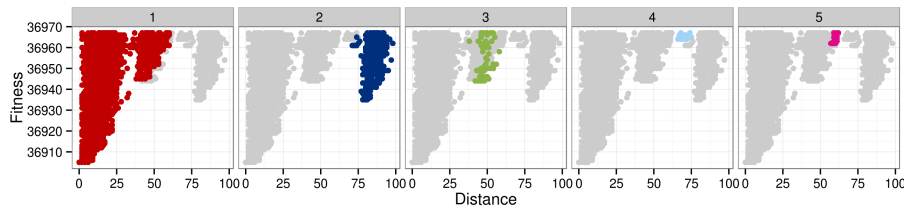
(b) u574

(c) rat575

(d) gr666

Fig. 5: Fitness-distance plots for the *best* 10 % sampling. All facets show the full set of solutions of the sampling in the background. Facets 1 to 5 display the overlay of the largest five connected components. Facet G shows the component containing the global optima (when it is not among the first five, as in the case of rat575). Facet R displays the remaining components if there are any.
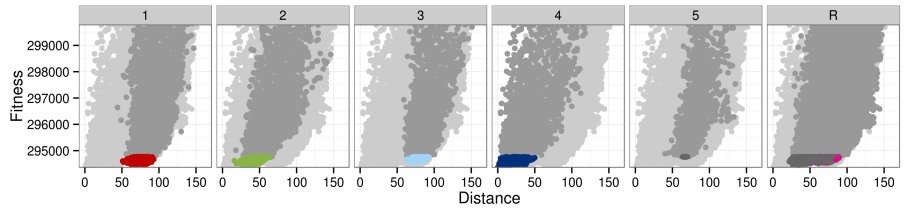
(a) att532
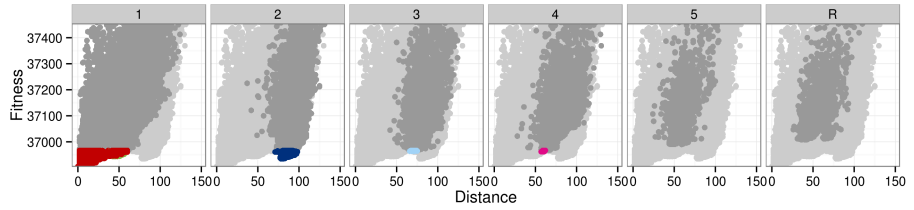
(b) u574
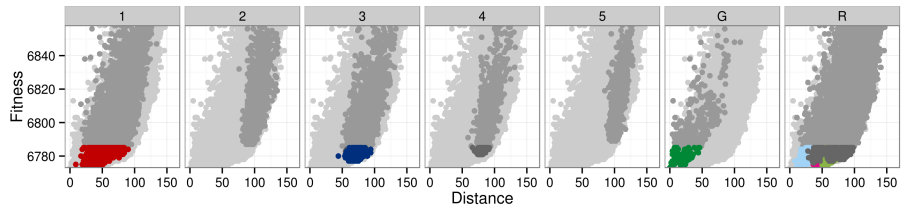
(c) rat575

(d) gr666

Fig. 6: Fitness distance plots for the *all* sampling. In the background (light grey), all facets show the set of sampled solutions below the 95th fitness percentile. Points within specific components are coloured in darker grey in each facet. Points in common with Figure 5 use the same colours. Facets 1 to 5 display the overlay of the largest five connected components. Facet G shows the component containing the global optima (when it is not among the first 5, such as in rat575). Facet R displays the remaining components.

best 10 % components are indeed at the bottom of a single massive component in the *all* sampling.

We can observe that there is relatively little overlap between components when the solution fitness is close to the best fitness (Fig. 5). This no longer the case when the *all* sampling is observed. The components not containing the global optima in instances att532 and u674, and to a lesser extent in gr666, are relatively far away from the one with the global optima both in terms of fitness and bond distance. In contrast, the best fitness values in the components of rat575 are all within 4 units of the global optimum. Let us note that, for this instance, the distribution of points appears to consist of distinct layers. This is simply because the range of fitness values is very small and all values are integers.

From Figure 6, it can be seen that the presence of distinct components does not match the big valley hypothesis, but rather that there are multiple distinct funnels. On some instances, looking at the bottom of these components, or funnels, reveals further splits into basins within funnels (Fig. 5).

6   Conclusions

Our study suggests that there is not always a single valley in the fitness landscape of travelling salesman problems under LK-search and double-bridge escape moves. Instead, local optima might decompose into a number of sub-valleys or funnels, as illustrated in Figure 7 for two funnels, but more than two are generally present. This decomposition occurs not only among solutions near in evaluation to the global optimum, but it may also happen among solutions with higher cost. In our local optima network model, the funnels are clearly identified and visualised as the connected components of the networks.

This has significant consequences in our understanding of iterated local search. Once the search process is trapped in a sub-optimal funnel, it simply cannot escape from it using the underlying escaping mechanism (double-bridge moves in our study). Increasing the number of iterations will not improve the performance, the search will stall, as transitions to other funnels are not possible. We foresee that this observation will inspire new escaping and tunnelling mechanisms that allow the search process to navigate among funnels.
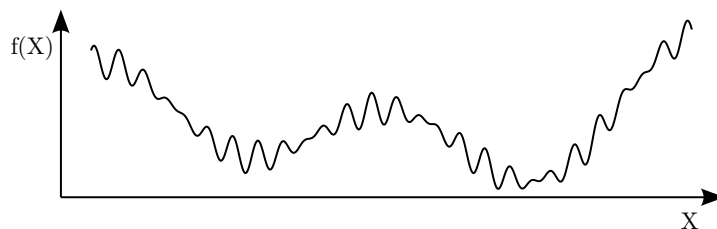

Fig. 7: Depiction of two funnels.

Future work will study the structure of larger, and more diverse TSP instances and other combinatorial problems where the big valley has been observed. More extensive sampling methods will need to be considered to confirm or infirm our results. We will also look at search strategies to escape from suboptimal funnels. We also aim to produce improved and informative images of fitness landscapes using the local optima network model.

Data Access. All data generated during this research are openly available from the Stirling Online Repository for Research Data (`http://hdl.handle.net/11667/71`).

References

1. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: Concorde TSP solver (2003), `http://www.math.uwaterloo.ca/tsp/concorde.html`
2. Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for Large Traveling Salesman Problems. INFORMS Journal on Computing 15, 82–92 (2003)
3. Boese, K.D., Kahng, A.B., Muddu, S.: A new adaptive multi-start technique for combinatorial global optimizations. Operations Research Letters 16, 101–113 (1994)
4. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal Complex Systems, 1695 (2006)
5. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software Practice Exper. 21(11), 1129–1164 (Nov 1991)
6. Hains, D.R., Whitley, L.D., Howe, A.E.: Revisiting the big valley search space structure in the TSP. Journal of the Operational Research Society 62(2), 305–312 (2011)
7. Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology 128, 11–45 (1987)
8. Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)
9. Klemm, K., Flamm, C., Stadler, P.F.: Funnels in energy landscapes. European Physical Journal B 63(3), 387–391 (2008)
10. Lin, S., Kernighan, B.W.: An Effective Heuristic Algorithm for the Traveling-Salesman Problem. Operations Research 21, 498–516 (1973)
11. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated Local Search. Handbook of Metaheuristics pp. 320–353 (2003)
12. Martin, O., Otto, S.W., Felten, E.W.: Large-step markov chains for the TSP incorporating local search heuristics. Operations Research Letters 11, 219–224 (1992)
13. Merz, P., Freisleben, B.: Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In: Proceedings of Parallel Problem Solving from Nature, PPSN V. Lecture Notes in Computer Science, vol. 1498, pp. 765–774. Springer-Verlag (1998)

14. Miller, M.A., Wales, D.J.: The double-funnel energy landscape of the 38-atom Lennard-Jones cluster. Journal of Chemical Physics 110(14) (1999)

15. Nešetřil, J., Milková, E., Nešetřilová, H.: Otakar Borůvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history. Discrete Mathematics 233(13), 3–36 (Apr 2001)

16. Newman, M.E.J.: Networks: An Introduction. Oxford University Press, Oxford, UK (2010)

17. Ochoa, G., Chicano, F., Tinos, R., Whitley, D.: Tunnelling crossover networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). pp. 449–456. ACM (2015)

18. Ochoa, G., Tomassini, M., Verel, S., Darabos, C.: A study of NK landscapes' basins and local optima networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). pp. 555–562. ACM (2008)

19. Ochoa, G., Veerapen, N., Whitley, D., Burke, E.: The Multi-Funnel Structure of TSP Fitness Landscapes: A Visual Exploration. In: Proceedings of Artificial Evolution, EA 2015. Lecture Notes in Computer Science, Springer (2015), to appear.

20. Ochoa, G., Verel, S., Daolio, F., Tomassini, M.: Local optima networks: A new model of combinatorial fitness landscapes. In: Richter, H., Engelbrecht, A. (eds.) Recent Advances in the Theory and Application of Fitness Landscapes, Emergence, Complexity and Computation, vol. 6, pp. 233–262. Springer Berlin Heidelberg (2014)

21. Reeves, C.R.: Landscapes, operators and heuristic search. Annals of Operations Research 86, 473–490 (1999)

22. Reinelt, G.: TSPLIB – A Traveling Salesman Problem Library. ORSA Journal on Computing 3(4), 376–384 (1991), `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

23. Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of NK landscapes with neutrality. IEEE Transactions on Evolutionary Computation 15(6), 783–797 (2011)

24. Whitley, D., Hains, D., Howe, A.: Tunneling Between Optima: Partition Crossover for the Traveling Salesman Problem. In: Proceedings Genetic and Evolutionary Computation Conference. pp. 915–922. GECCO '09, ACM, New York, NY, USA (2009)