Tunnelling Crossover Networks
for the Asymmetric TSP

Nadarajen Veerapen[1(✉)], Gabriela Ochoa[1], Renato Tinós[2],
and Darrell Whitley[3]

[1] Division of Computing Science and Mathematics,
University of Stirling, Stirling, UK
`nve@cs.stir.ac.uk`
[2] Department of Computing and Mathematics,
University of São Paulo, São Paulo, Brazil
[3] Department of Computer Science, Colorado State University, Fort Collins, USA

Abstract. Local optima networks are a compact representation of fitness landscapes that can be used for analysis and visualisation. This paper provides the first analysis of the Asymmetric Travelling Salesman Problem using local optima networks. These are generated by sampling the search space by recording the progress of an existing evolutionary algorithm based on the Generalised Asymmetric Partition Crossover. They are compared to networks sampled through the Chained Lin-Kernighan heuristic across 25 instances. Structural differences and similarities are identified, as well as examples where crossover smooths the landscape.

1 Introduction

The global structure of fitness landscapes in combinatorial optimisation is far from being well-understood, and yet crucially impacts the dynamic of search heuristics. The operators within such algorithms usually restrict the search space in some way, potentially over-exploring or missing key parts of the actual landscape. Tools to better understand and visualise fitness landscapes are therefore needed. The symmetric Travelling Salesman Problem (TSP) has been widely studied. Its more general formulation, the Asymmetric TSP (ATSP) has received less attention but is useful to model real-world situations where symmetry is often a luxury. In this paper, we attempt to provide some insights into its landscape structure by studying local optima networks.

Local optima networks (LON) are graph-based models of combinatorial fitness landscapes, originally inspired by work on energy landscapes in computational chemistry [4]. A fitness landscape is compressed into a graph where nodes are local optima and edges possible search transitions among them [10,17]. The first model considered binary search spaces and the NK family of landscapes; nodes were local optima according to a best-improvement local search with bit-flip moves, and edges account for transition probabilities among basins of attraction [10]. This model required a full enumeration of local optima and basins, and was therefore impossible to scale to realistically sized landscapes. An alternative

definition of edges was later proposed to account for *escape* probabilities among optima, that is, probabilities to hop from a local optimum to another after a perturbation (large mutation) followed by local search [18]. Recently, sampling approaches have been developed using escape edges in order to model landscapes of realistic size [6,11–13]. In particular, work on the symmetric Travelling Salesman Problem has revealed intriguing landscape visualisations, providing compelling evidence of the existence of multiple valleys or clusters of local optima (also called funnels) on the studied instances [12,13]. Most local optima network models so far consider transitions based on perturbation operators. Ochoa et al. [9] proposed a model where transitions are based on recombination. Specifically, the deterministic Partition (Tunnelling) Crossover by Tinós et al. was considered [16], together with efficient procedures for extracting all the local optima of NK landscapes of string length up to 30, based on exploiting the structure of pseudo-Boolean problems with bounded epistasis [3].

The main goal of this article is to model tunnelling crossover networks for asymmetric Travelling Salesman instances of realistic size. More specifically, the contributions are:

1. First study of local optima networks for the asymmetric TSP.
2. An extension of the local optima network model to capture evolutionary algorithms. This is achieved by incorporating two types of edges, one based on mutation and another based on recombination.
3. A network sampling mechanism based on instrumenting an existing evolutionary algorithm.
4. Comparing the local optima network structure emerging from an evolutionary algorithm against a single-point heuristic (Iterated Local Search).

Following this introduction, the paper presents the crossover operator in Sect. 2. Section 3 provides key definitions for local optima networks and describes how the network data are gathered. Section 4 presents the instances, which are analysed in Sect. 5. The conclusion is found in Sect. 6.

2   Generalised Asymmetric Partition Crossover

Our study considers the Generalised Asymmetric Partition Crossover (GAPX), a deterministic recombination operator proposed by Tinós et al. [15] for the Asymmetric Travelling Salesman Problem. GAPX is based on the Generalised Partition Crossover (GPX), developed by Whitley et al. [19] for the symmetric TSP. GAPX and GPX recombine partial solutions that are not shared in common between two parent solutions. First, a union graph $G_u = G_1 \cup G_2$ is created from graphs G_1 and G_2 representing the parent solutions. Then, common edges are removed from G_u and connected components are identified. Some of the connected components are the *recombining components*, i.e., connected subgraphs that can be deterministically recombined. GAPX and GPX find the best recombinations among the recombining components in order to generate the offspring. If the number of recombining components is q, then the best of 2^q
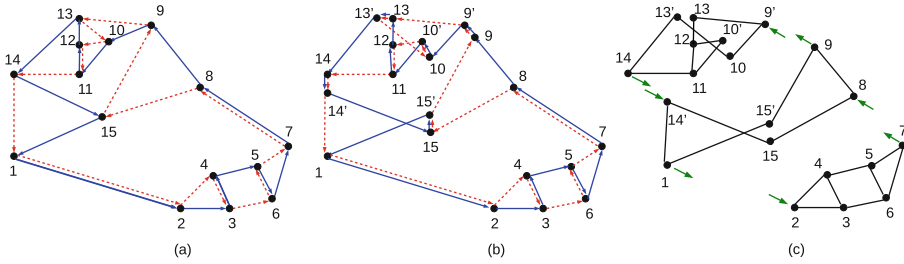
Fig. 1. Recombining two parent solutions using GAPX. (a) Parent solutions are shown by solid (blue) and dashed (red) lines. (b) In a first step, ghost vertices are inserted after vertices of degree 4. (c) Common edges are removed, allowing to identify 3 partitions. (Color figure online)

offspring is found at computational cost $O(n)$. This is possible because the partial evaluations of each one of the q recombining components are independently computed.

In GPX, the recombining components are the connected components separated from the rest of the graph by exactly two common edges. The remainder of the graph is also a recombining component. The Lin-Kernighan-Helsgaun (LKH) algorithm [5] includes a recombination operator, called Iterative Partial Transcription (IPT), which is similar in effect to GPX.

The GAPX includes enhancements to GPX that allow it to find many more recombining partitions than GPX and IPT. As a consequence, an exponentially higher number of offspring is explored. For example, when recombining the two parents shown in Fig. 1a, GPX (adapted to the asymmetric TSP) finds $q = 2$ partitions, while GAPX finds $q = 3$ partitions. Thus, while GPX finds the best of $2^2 = 4$ offspring in this example, GAPX finds the best of $2^3 = 8$ offspring.

One enhancement to GPX is that GAPX exploits cuts that break nodes of degree 4 of G_u as a site for crossover. This is possible by splitting every vertex of degree 4 in order to create "ghost" vertices (Fig. 1b). According to the the direction of flow given by the solutions, common edges between the original vertices and their respective ghost vertices can be created. Such common edges are candidate sites for crossover when the connected components of the new union graph are identified.

3   Local Optima Networks for TSP

Nodes and edges make up the networks. They are defined by the methodology for extracting the network data which is described in the next subsection. A full enumeration of the local optima for ATSP instances of non-trivial size is clearly 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 none of its neighbours is shorter than it. The set of local optima is denoted by *LO*.

The neighbourhood is imposed by k-opt local search. The later is applied after a crossover or mutation operation, in the case of an evolutionary algorithm, and after a perturbation, in the case of an Iterated Local Search. A k-opt local search considers all the possibilities of exchanging k edges in a tour and picks the best. The local optimality criterion is, therefore, rather stringent since only a small number of tours are k-optimal.

Definition 2. *Edges* are directed and of different types based on crossover, mutation, or perturbation. There is an edge from local optimum LO_i to local optimum LO_j, if LO_j can be obtained after any of those operations to LO_i followed by k-opt search. The set of edges is denoted by E.

In the case of the crossover, a pair of edges is created: one starting from each parent and targeting the offspring as has been done in [9]. However, the local optima networks in the latter did not include a second type of edge based on a mutation operator, which we include here.

Definition 3. The *local optima network, LON*, is the graph $LON = (LO, E)$ where nodes are the local optima *LO*, and set E are the edges.

3.2   Gathering Network Data

The GAPX network data is generated by instrumenting and adapting the genetic algorithm from Tinós et al. [15] (see Algorithm 1). After each crossover or mutation operation, the solution obtained is transformed using 3-opt and each unique local optimum obtained is stored in *LO*. We also store, in E, an edge between the starting and end optima after one of these two operations. If no improving solution is found during 20 consecutive generations, all the solutions in the population, except the ones with best fitness, are replaced by random solutions improved by 3-opt.

Contrary to the GA from [15], a full 3-opt is performed, not a greedy version. This is done after all crossover and mutation operations. The algorithm can therefore be described as a fully hybrid algorithm, combining a GA and local search. The mutation operator consists of a sequence of up to 5 double-bridge moves, i.e., exchanges of 4 edges in a specific pattern. The algorithm is run 100 times with a population of 100 individuals, until a global optimum is found or 500 generations have elapsed. These parameters also depart from the original ones (300 individuals and 1500 generations), otherwise success rates of 100 % in finding a global optimum are observed on most instances. Discriminating between easier and harder instances would therefore be more difficult.

To provide a basis for the comparison of the GAPX network data, we use network data from an Iterated Local Search based on the well-known Lin-Kernighan

$P \leftarrow \text{popInit}()$
while *termination condition is not satisfied* **do**
    $Q(1) \leftarrow \text{bestSolution}(P)$
    for $i \leftarrow 2$ **to** *maxPop* **do**
        $(p_1, p_2) \leftarrow \text{selection}(P)$
        $Q(i) \leftarrow \text{3opt}(\text{crossover}(p_1, p_2))$
        $LO \leftarrow LO \cup \{Q(i)\}; E \leftarrow E \cup \{(p_1, Q(i)), (p_2, Q(i))\}$
        if *crossover did not improve the solutions* **then**
            $best \leftarrow \text{chooseBest}(p_1, p_2)$
            $Q(i) \leftarrow \text{3opt}(\text{doubleBridgeMutation}(best))$
            $LO \leftarrow LO \cup \{Q(i)\}; E \leftarrow E \cup \{(best, Q(i))\}$
        end
    end
    if *best sol. did not improve in last 20 gen.* **then** $Q \leftarrow \text{immigration}(P)$
    $P \leftarrow Q$
end

Algorithm 1. Local optima network sampling in evolutionary algorithm.

(LK) heuristic [8]. We instrumented [13] the Chained Lin-Kernighan (Chained-LK) implementation by Applegate et al. [2] provided in the Concorde TSP solver [1]. LK applies 2, 3 and higher-order k-opt moves, with k chosen adaptively. The perturbation operator in Chained-LK is a double-bridge. Let us note that LK is designed for the symmetric TSP. A conversion step is required to handle ATSP instances. The process is described in the next section.

Ensuring fairness between two different algorithms when gathering local optima is not obvious. We chose to first run the hybrid algorithm and record the total number of edges that had been travelled across the 100 runs for each instance. Chained-LK was then executed such that it performed enough runs to have travelled across as many edges. Each of these runs ends when a global optimum has been found or when $20n$ perturbations have been performed, with n being the number of vertices in the original ATSP instance. On the instances where this method does not lead to at least 100 Chained-LK runs, additional runs are executed to reach 100.

4   Selected ATSP Instances

Our study considers the ATSP instances from TSPLIB [14] belonging to different types, as well as instances generated using the DIMACS symmetric TSP generator code. We use two types of generated instances: uniformly distributed cities (prefixed by E) and clustered cities (prefixed by C). The symmetric instances are transformed into asymmetric instances by inserting random Gaussian deviations (with standard deviation equal to $0.2d_{i,j}$) to each distance $d_{i,j}$, where i and j are any two cities. By considering this variety of instances, our aim is to discover structural differences distinguishing the hard from the easy to solve

instances. The instances from TSPLIB are mostly real-life instances, originating from sequencing, scheduling, vehicle routing and stacker crane problems. Instances *ry48p* and *kro124p* are synthetic.

The Concorde exact solver was used to compute the minimal fitness for the generated instances. Since Concorde can only handle symmetric TSP instances, the ATSP instances were transformed into symmetric instances by doubling the number of vertices [7]. Given a set V of n vertices and the distance $d_{i,j}, \forall i, j \in V$, a new city $n + i$ is created $\forall i \in V$. The cost of edge $(i, n + i)$ is set to $0, \forall i \in V$, the cost of $(n + i, j)$ is set to $d_{i,j} + M, \forall i, j \in V$, where M is a sufficiently large number, and the cost of the remaining edges is set to ∞. The value nM is subtracted from the fitness. The same transformation was used to convert the ATSP instances into instances that are suitable for Concorde's Chained-LK heuristic.

5   Network Analysis

The execution and network data generated are summarised in Table 1. *GAPX* indicates results for the hybrid algorithm based on the GAPX. *CLK* indicates results based on Chained-LK. The column titled *Runs* indicates the number of Chained-LK runs required to traverse at least as many edges as where traversed by 100 runs of the GAPX-based algorithm. *Success* represents the proportion of runs that find a global optimum. *Unique Opt.* refers to the number of unique global optima. *Conn. Comp* refers to the number of connected components. *Edge Opt.* and *Mut.* show the proportion of GAPX edges where the end node is already 3-opt before local search and the proportion of edges that are mutation edges, respectively.

The first observation is that at least one global optimum has been found on most instances for each solving method. This indicates that, although the algorithms and their parameters may not be perfect, the best solutions are reachable using these two sampling approaches. We may therefore interpret the results with a minimal level of confidence that they represent a non-trivial part of the landscape. Furthermore, with the chosen parameters, Chained-LK sometimes has similar success rates to the GAPX-based algorithm on several instances, which would tend to show that the *edge budget* allocated is sufficient to fairly compare the two types of networks.

The smallest and easiest instance, br17, exhibits a smooth landscape under the different operators used. Global optima are found in the first generation or iteration, which is highlighted by the high number of connected components. This is an artefact of the sampling algorithms which terminates a run as soon as a global optimum is found. Thus plateaus of global optima are not fully explored.

At the opposite end, the largest instances (rbg) are very easy for the GAPX-based algorithm. Chained-LK, on the other hand, struggles and its 100 separate runs end up in 100 different funnels (connected components). However, the mean fitness of the Chained-LK nodes is lower than that of GAPX nodes. This showcases GAPX's ability to *tunnel through* what is a totally different landscape for

Table 1. Network data. *Success* – proportion of runs that find a global optimum. *Edge Opt.* and *Mut.* – proportion of GAPX edges where the end node is already 3-opt before local search and the proportion of edges that are mutation edges, respectively.

	Inst.	Size	Runs	Success		Unique Opt.		Conn. Comp.		Nodes		Edges		Edge Opt.	Mut.
			CLK	GAPX	CLK	GAPX	CLK	GAPX	CLK	GAPX	CLK	GAPX	CLK	GAPX	GAPX
1	br17	17	28915	1.00	1.00	4843	23669	1966	10726	20332	25722	24821	17001	0.98	0.38
2	ftv33	34	1463	1.00	0.99	2	2	110	1	46854	2285	90360	6202	0.70	0.25
3	ftv35	36	3242	1.00	0.57	1	1	72	1	287402	6384	850330	13984	0.67	0.31
4	ftv38	39	3592	0.98	0.43	2	2	16	1	685874	9971	2125869	20918	0.67	0.31
5	p43	43	7766	0.87	0.47	98	3685	247	3439	2221484	988626	4464259	1340720	0.65	0.27
6	ftv44	45	891	1.00	0.64	16	16	64	1	131789	3959	298787	7905	0.68	0.28
7	ftv47	48	1615	1.00	0.55	16	16	10	1	287511	8853	701515	17052	0.68	0.28
8	ry48p	48	3035	1.00	0.33	1	1	24	1	644854	16212	1669819	28629	0.66	0.30
9	C50.0	50	9604	1.00	0.95	2	2	21	1	505885	28429	1314569	53336	0.65	0.32
10	E50.0	50	10069	1.00	1.00	2	2	50	1	121457	21012	255856	40388	0.70	0.29
11	ft53	53	932	1.00	0.27	1	1	4	1	214726	6611	555520	9784	0.64	0.32
12	ftv55	56	4622	1.00	1.00	85	128	121	1	208918	30718	443094	65671	0.70	0.27
13	ftv64	65	1683	1.00	0.71	95	132	31	4	341157	22646	748388	47802	0.70	0.26
14	ft70	70	7461	0.58	0.23	1	1	32	1	2496591	86694	6648585	132447	0.56	0.30
15	ftv70	71	4897	1.00	0.96	82	128	20	1	527355	61919	1185003	139180	0.71	0.26
16	kro124p	100	3109	1.00	0.86	1	1	34	1	597158	33167	1370484	52971	0.62	0.31
17	C100.0	100	4391	0.62	0.12	2	2	105	26	2509747	62616	5394010	96362	0.62	0.31
18	E100.0	100	614	1.00	0.38	2	2	48	2	470359	7602	921047	11930	0.64	0.32
19	ftv170	171	3357	0.55	0.36	79	1206	131	599	3310144	779809	6835402	1516150	0.68	0.26
20	C200.0	200	3044	0.06	0.02	1	2	166	877	4127102	117804	8171590	183914	0.62	0.28
21	E200.0	200	3070	0.02	0.01	1	2	135	894	4140274	97048	8085149	152355	0.61	0.30
22	rbg323	323	100	1.00	0.00	589	0	1868	100	20773	252775	22840	258353	0.20	0.11
23	rbg358	358	100	1.00	0.00	470	0	1939	100	20807	311891	22862	317813	0.17	0.10
24	rbg403	403	100	1.00	0.04	1418	4	2073	100	18241	534931	18904	540314	0.19	0.13
25	rbg443	443	100	1.00	0.22	1909	22	2091	100	18498	513508	18905	518744	0.12	0.09

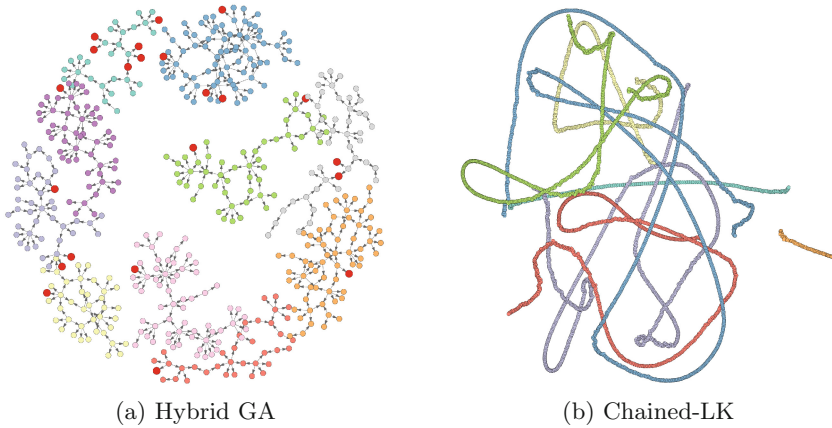(a) Hybrid GA   (b) Chained-LK

Fig. 2. Subsets of local optima networks for rbg323 under a fitness threshold of 1331. The hybrid GA network is further simplified by selecting only the 10 largest components (513 nodes). The global optima, with fitness 1326, are painted red. The Chained-LK network contains 3883 nodes in 7 components indicated by different colours, the smaller one on a plateau of fitness 1329, the others on a plateau of fitness 1331. (Color figure online)

Chained-LK. It is interesting to note that, as opposed to other instances, the majority of nodes generated through crossover are not 3-optimal and that there is a low proportion of mutation edges as well. GAPX combined with 3-opt are thus able to drive the search through the landscape largely without mutation. Figure 2 shows a subset of the local optima networks for rbg323 very close to the global optima. The structure difference is striking, with the Chained-LK network stuck on two plateaus.

Connected-component-wise, Chained-LK has a tendency to generate landscapes with fewer components while finding more unique global optima in general. This may seem surprising given that Chained-LK recorded fewer unique nodes and edges. These numbers are smaller since local optima that do not improve the current solution in a run are not recorded due to memory constraints. Otherwise these worsening solutions would make up the majority of nodes.

Pearson's correlation coefficients were computed for pairwise comparisons of several execution and landscape features. One of them is the mean normalised fitness of nodes, not displayed in Table 1, which is always under 0.3 units. It is negatively correlated to the number of edges (-0.54) and nodes (-0.60) for Chained-LK but there is no correlation for GAPX. It is the opposite for the number of connected components, which is not correlated to the mean normalised fitness for Chained-LK but shows a negative correlation (-0.65) for GAPX.

The number of connected components is strongly correlated (1.0) to the number of unique local optima for Chained-LK and to a lesser degree (0.72) for GAPX. In the context of crossover and mutation networks, the proportion

of crossover edges that end in a local optima before the application of 3-opt is strongly correlated (0.92) to the proportion of mutation edges. This indicates that mutation is usually required to increase diversity.

This work is a first attempt at extracting crossover networks of moderately large permutation problem instances. As such it has some limitations. For example, the nodes' mean in-degree across instances ranges between 1 and 3. This is different from the results obtained by [9], where the mean in-degree ranged from 0.8 to 245 for exhaustively sampled NK landscapes. This could simply be due to the nature of ATSP landscapes, the use of mutation edges or be the result of some bias in the sampling. Further work will investigate such issues.

6   Conclusion

Local optima networks help to better understand the global structure of combinatorial landscapes by providing a relatively compact representation. Nevertheless, sampling is required to study instances on non-trivial sizes. We have done this here for the Asymmetric TSP with networks generated from an evolutionary and an iterated local search algorithm. We have presented evidence of their differences and similarities. On some larger real-life instances the crossover-based algorithm produced networks that were drastically different from the other approach, effectively demonstrating the tunnelling behaviour of carefully designed crossover operators.

This work only scratches the surface of the use of local optima networks for understanding evolutionary algorithms and the structure of non-trivial combinatorial problem instances. Further work will look at improving the sampling, both in the methodology and the quantity of data points gathered. We also intend to carry out systematic investigations of a wide range of instances with different characteristics.

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 J. Comput. **15**, 82–92 (2003)
3. Chicano, F., Whitley, D., Sutton, A.M.: Efficient identification of improving moves in a ball for pseudo-boolean problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014), pp. 437–444. ACM (2014)

4. Doye, J.P.K.: The network topology of a potential energy landscape: a static scale-free network. Phys. Rev. Lett. **88**, 238701 (2002)
5. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. Eur. J. Oper. Res. **126**(1), 106–130 (2000)
6. Iclanzan, D., Daolio, F., Tomassini, M.: Data-driven local optima network characterization of QAPLIB instances. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014), pp. 453–460. ACM (2014)
7. Jonker, R., Volgenant, T.: Transforming asymmetric into symmetric traveling salesman problems. Oper. Res. Lett. **2**(4), 161–163 (1983)
8. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. **21**, 498–516 (1973)
9. Ochoa, G., Chicano, F., Tinós, R., Whitley, D.: Tunnelling crossover networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015), pp. 449–456. ACM (2015)
10. 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 2008), pp. 555–562. ACM (2008)
11. Ochoa, G., Veerapen, N.: Additional dimensions to the study of funnels in combinatorial landscapes. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016). ACM (2016, to appear)
12. Ochoa, G., Veerapen, N., Whitley, D., Burke, E.K.: The multi-funnel structure of TSP fitness landscapes: a visual exploration. In: Bonnevay, S., Legrand, P., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) EA 2015. LNCS, vol. 9554, pp. 1–13. Springer, Heidelberg (2016). doi:10.1007/978-3-319-31471-6_1
13. Ochoa, G., Veerapen, N.: Deconstructing the big valley search space hypothesis. In: Chicano, F., Hu, B., García-Sánchez, P. (eds.) EvoCOP 2016. LNCS, vol. 9595, pp. 58–73. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30698-8_5
14. Reinelt, G.: TSPLIB - a traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991)
15. Tinós, R., Whitley, D., Ochoa, G.: Generalized asymmetric partition crossover (GAPX) for the asymmetric TSP. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014), pp. 501–508. ACM (2014)
16. Tinós, R., Whitley, L.D., Chicano, F.: Partition crossover for pseudo-boolean optimization. In: Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, 17–20 January 2015, pp. 137–149 (2015)
17. Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of NK landscapes with neutrality. IEEE Trans. Evol. Comput. **15**(6), 783–797 (2011)
18. Vérel, S., Daolio, F., Ochoa, G., Tomassini, M.: Local optima networks with escape edges. In: Hao, J.-K., Legrand, P., Collet, P., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) EA 2011. LNCS, vol. 7401, pp. 49–60. Springer, Heidelberg (2012)
19. Whitley, D., Hains, D., Howe, A.: Tunneling between optima: partition crossover for the traveling salesman problem. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2009), pp. 915–922. ACM (2009)